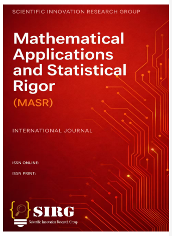


Mathematical Applications and Statistical Rigor
MASR

Journal Homepage
https://pub.scientificirg.com/index.php/MASR/index

Advances in Uniform Experimental Designs: A Decade Selective Review of Algorithmic Search and Deterministic Construction Methods

S.M. Celem[a,1] , **Laala Barkahoum**[b], **Gajendra K. Vishwakarma**[c] **and Hong Qin**[a,d]

[a] *School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan 430073, China*
[b] *Department of Mathematics, Faculty of Exact Sciences, Constantine 1 Mentouri Brothers University, Algeria*
[c] *Department of Mathematics & Computing, Indian Institute of Technology Dhanbad, Dhanbad, India*
[d] *Faculty of Mathematics and Statistics, Central China Normal University, Wuhan 430079, China*

ABSTRACT

This paper presents a comprehensive and selective review of the last decade's progress in the construction of uniform experimental designs. Confronting the growing complexity of modern experiments—characterized by high-dimensional factor spaces and constrained resources—recent research has produced promising methods tools for constructing efficient, cost-effective designs. The review is organized around three pivotal themes: (1) enhanced stochastic search algorithms, including adjusted threshold accepting and permutation-projection methods, for constructing (nearly) uniform minimum aberration designs, supported by benchmarks that reduce computational search; (2) frameworks for constructing uniform fold-over designs in two-stage sequential experimentations, enabling the breaking of aliasing structures across symmetric and asymmetric designs; and (3) deterministic construction algorithms—such as multiple doubling, tripling, and quadrupling, and their integration—for efficiently generating large-scale uniform designs with low, high, or mixed levels without exhaustive search. Collectively, these advances offer researchers a robust, computationally efficient, and theoretically coherent toolkit for designing experiments across scientific and industrial domains, representing a substantial leap beyond conventional methodologies. A critical discussion and comparative analysis of the reviewed methods are also provided, along with practical recommendations for implementation.

PAPER INFORMATION

HISTORY
Received: 25 August 2025
Revised: 15 November 2025
Accepted: 10 January 2026
Online: 20 January 2026

MSC
62K05
62K15

KEYWORDS
Uniform designs
Fold-over designs
Algorithmic construction
Design optimality criteria
Fractional factorial designs
Minimum aberration designs

1 Introduction

Experimentation is a fundamental activity across countless domains, from everyday problem-solving to advanced technological innovation in fields such as materials science, engineering, semiconductors, robotics, and life sciences. Whether in formal scientific research, industrial development, or general practice, investigators rely on experiments to understand complex phenomena. As scientific and technological systems have advanced, the systems under study have grown increasingly complex, often involving numerous inputs, or *factors*. Each factor may have two or more discrete settings, known as *levels*. Understanding how changes in these levels affect a system's output, or *response*, is essential; yet

[1]Corresponding author at School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan 430073, China.
E-mail: smcelem@yahoo.com.

the underlying relationships among factors are seldom straightforward. In such contexts, conclusions drawn solely from intuition and experience prove insufficient. Consequently, investigators across virtually every field have come to recognize the critical importance of rigorous experimental design.

A well-structured design enables a deeper understanding of how input factors influence system performance. This knowledge, in turn, guides experimenters in identifying the active factors that significantly affect the system and in selecting optimal factor levels to improve desired outputs. For example, in industrial settings, key objectives such as enhancing product quality, increasing productivity, and improving resource efficiency can be systematically pursued through experimental design. Given that production lines often operate for decades, carefully designed experiments offer a powerful methodology for optimizing established processes. Common goals in industrial engineering and scientific practice include increasing process yields, improving product quality and reliability, reducing development time, lowering costs, evaluating material alternatives, selecting design parameters, and modeling relationships between inputs and outputs.

Both physical and computer simulations have become indispensable tools in science, engineering, and industry, with recent applications documented in medicine [1], civil engineering [2], materials science [3], biomedical engineering [4], agriculture [5], clinical trials [6], and rocketry [7]. An experiment involving more than one factor is termed a *factorial experiment*, and the strategic selection of its level combinations, or *runs*, constitutes a *factorial design*. Factorial designs are widely employed to study several-factors-at-a-time (SFAT) simultaneously, each at multiple discrete levels. Empirical evidence consistently shows that the SFAT approach is more efficient than varying one-factor-at-a-time (OFAT). Compared to OFAT experiments, factorial designs offer several key advantages: they yield more comprehensive information at a similar or lower cost; they facilitate faster convergence to optimal conditions; and, crucially, they can detect interactions between factors—effects where the influence of one factor depends on the level of another—which are entirely invisible to the OFAT methodology.

Space-filling designs have emerged as a powerful paradigm for this purpose, aiming to distribute a limited number of points uniformly throughout the experimental region [8, 9]. Among these, uniform designs [10] have gained prominence due to their robustness and excellent performance [11, 12], particularly when the underlying input-output relationship is unknown or highly nonlinear [13]. Their ability to provide good coverage of the design space with relatively few samples makes them exceptionally valuable for exploring complex computer models [14, 15].

This paper presents a selective review of the most significant algorithmic and theoretical advances in uniform experimental design over the past decade. Confronted with challenges such as high-dimensional factor spaces, limited runs, and mixed-level structures, researchers have developed transformative tools for constructing highly efficient, cost-effective designs. We organize these contributions around three interconnected themes: stochastic search algorithms, fold-over frameworks for sequential experimentation, and deterministic construction methods. Collectively, these advances offer a robust, computationally efficient toolkit that represents a substantial leap beyond conventional design methodologies. The remainder of the paper is structured as follows: Section 2 introduces foundational frameworks, notations, and optimality criteria; Section 3 reviews enhanced stochastic search algorithms; Section 4 covers fold-over frameworks; Section 5 details deterministic construction methods; and Section 6 concludes with a critical discussion and practical recommendations.

2 Frameworks, Notations, and Criteria

2.1 Full Factorial Designs (FuFDs)

The complexity of real-world processes, often combined with limited prior insight, makes the collection of informative and efficient datasets—factorial designs—a significant challenge. These designs are crucial for accurately estimating factor effects and understanding system behavior. A *full factorial design (FuFD)* is one in which experiments are conducted at every possible combination of levels for all active factors. Formally, for s factors where the k-th factor has q_k discrete levels, a FuFD consists of all $\prod_{k=1}^{s} q_k$ level combinations. This exhaustive approach allows experimenters to study a system under all possible conditions, enabling the estimation of all main effects (individual factor influences) and interaction effects (joint influences of multiple factors).

A Framework Based on Linear Regression: Consider an experiment investigating the relationship between a quantitative response Y and s quantitative factors $F_1, \dots, F_s$, where each factor F_k varies within an interval $[a_k, b_k]$. The experimenter selects n runs, $\mathbf{r}_1, \dots, \mathbf{r}_n$, from the experimental domain $\mathcal{D} = [a_1, b_1] \times \dots \times [a_s, b_s]$ and observes the corresponding responses $y_1, \dots, y_n$. Each run is a vector $\mathbf{r}_i = (f_{i,1}, \dots, f_{i,s})$, with $f_{i,k} \in [a_k, b_k]$. A linear regression model incorporating all main effects and interactions up to order s is:

$$Y = \beta_0 + \sum_{k=1}^{s} \beta_k F_k + \sum_{1 \le j < k \le s} \beta_{j,k} F_j F_k + \dots + \beta_{1,2,\dots,s} \prod_{k=1}^{s} F_k + \varepsilon, \tag{1}$$

where $\beta_0, \beta_k, \beta_{j,k}, \dots$ are regression coefficients, and ε is a random error with $E(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$. The model can be expressed in matrix form as:

$$\mathbf{Y} = \mathbf{A}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2}$$

where $\mathbf{Y} = (y_1, \ldots, y_n)^\top$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$, $\boldsymbol{\beta}$ is the vector of all coefficients, and $\mathbf{A} = [\mathbf{1}_n, \mathbf{B}]$ is the model matrix [16]. Here, $\mathbf{1}_n$ is an n-vector of ones, and $\mathbf{B}$ is the matrix containing all main effect and interaction columns (e.g., $F_1, F_1F_2, \ldots$). The *design matrix* $\mathbf{d}$ contains only the columns for the factor settings used in the runs. The ordinary least squares estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{Y}. \tag{3}$$

To uniquely estimate all parameters, the matrix $\mathbf{A}^\top\mathbf{A}$ must be nonsingular, which requires $\mathbf{A}$ to be of full column rank. A FuFD, by its exhaustive nature, guarantees this full rank condition when all effects in the model are estimable.

Example 2.1 *A 2^3 FuFD: Consider a FuFD with three factors, each at two levels coded as -1 (low) and $+1$ (high). This 2^3 design is presented in Table 1. Key features of this design are: The total number of runs is $2^3 = 8$, covering all possible level combinations. The columns of the design matrix $\mathbf{d} = [F_1, F_2, F_3]$ are orthogonal. The experiment is performed at each run to obtain the response vector $\mathbf{Y} = (y_1, \ldots, y_8)^\top$. Replicates may be used, in which case the average response per run can be analyzed. The eight columns of the full model matrix $\mathbf{A} = [\mathbf{1}_8, F_1, F_2, F_3, F_1F_2, F_1F_3, F_2F_3, F_1F_2F_3]$ are linearly independent, ensuring $(\mathbf{A}^\top\mathbf{A})$ is nonsingular. This allows for the unique estimation of all eight model parameters (β_0, three main effects, three two-factor interactions, and one three-factor interaction) by solving the system of eight equations derived from Model (1). The model in Equation (1) uses coded factor levels (e.g., -1 and $+1$). The relationship between a coded value F_{code} and its actual value F_{actual} within a range $[F_{min}, F_{max}]$ is given by $F_{code} = \frac{F_{actual} - \frac{1}{2}(F_{max}+F_{min})}{\frac{1}{2}(F_{max}-F_{min})}$.*

Table 1: The 2^3 FuFD and the corresponding expanded model matrix $\mathbf{A}$.

	$\mathbf{A} = [\mathbf{1}_8\ F_1\ F_2\ F_3\ F_1F_2\ F_1F_3\ F_2F_3\ F_1F_2F_3]$								
	$\mathbf{B} = [F_1\ F_2\ F_3\ F_1F_2\ F_1F_3\ F_2F_3\ F_1F_2F_3]$								
	$\mathbf{d} = [F_1\ F_2\ F_3]$			Output					
Run	F_1	F_2	F_3	$\mathbf{Y}$	F_1F_2	F_1F_3	F_2F_3	$F_1F_2F_3$	$\mathbf{1}_8$
r_1	-1	-1	-1	y_1	$+1$	$+1$	$+1$	-1	$+1$
r_2	-1	-1	$+1$	y_2	$+1$	-1	-1	$+1$	$+1$
r_3	-1	$+1$	-1	y_3	-1	$+1$	-1	$+1$	$+1$
r_4	-1	$+1$	$+1$	y_4	-1	-1	$+1$	-1	$+1$
r_5	$+1$	$+1$	-1	y_5	$+1$	-1	-1	-1	$+1$
r_6	$+1$	$+1$	$+1$	y_6	$+1$	$+1$	$+1$	$+1$	$+1$
r_7	$+1$	-1	-1	y_7	-1	-1	$+1$	$+1$	$+1$
r_8	$+1$	-1	$+1$	y_8	-1	$+1$	-1	-1	$+1$
Coefficient	β_1	β_2	β_3		$\beta_{1,2}$	$\beta_{1,3}$	$\beta_{2,3}$	$\beta_{1,2,3}$	β_0

2.2 Fractional Factorial Designs (FrFDs)

Conducting a FuFD for experiments with many factors is often prohibitively expensive or time-consuming, especially for real-world physical trials. Even in computer simulation, where per-run costs are lower, evaluating large, complex models can demand significant computational resources. Therefore, selecting an informative subset of experimental points—a *fractional factorial design (FrFD)*—is a critical practical solution. A well-chosen FrFD maximizes valuable information about system behavior while minimizing the number of runs. Formally, a FrFD selects a fraction (e.g., $1/2$, $1/4$) of the runs from a corresponding FuFD. The central challenge is to select this fraction efficiently to estimate the most important effects with a limited budget. FrFDs are broadly classified into two types: *regular* and *non-regular*.

Regular FrFDs are constructed through *defining relations* among factors, which leads to a structured *aliasing* pattern. In these designs, any two effects are either orthogonal (estimable independently) or fully aliased (completely confounded, making only their linear combination estimable). This simple aliasing structure has made regular FrFDs a mainstay in both theory and application. Regular FrFDs have been extensively studied and applied due to their mathematical simplicity and well-understood properties (e.g., [17, 18]). However, a key limitation is that their run size n must be a power of the number of levels q (i.e., $n = q^k$ for some integer k). This constraint can lead to large gaps between allowable run sizes, especially when q or k is large. Non-regular FrFDs offer greater flexibility. They are not based on defining relations and can be constructed for a wider variety of run sizes, effectively filling the gaps left by regular designs. This makes them highly adaptable to practical constraints. The trade-off for this flexibility is a more complex aliasing structure. In non-regular designs, effects are typically *partially aliased* rather than fully aliased. This means main effects may be partially correlated with some interaction effects, making the statistical analysis more challenging. However, research has shown that from certain estimation efficiency perspectives, non-regular designs can outperform their regular counterparts, which has driven significant recent interest in their development and application (e.g., [19]).

Example 2.2 *A 2^{3-1} FrFD: Consider an experiment with three two-level factors. A full factorial 2^3 design requires 8 runs. If resources are limited to only 4 runs, a half-fraction (2^{3-1}) can be constructed. This is achieved by treating F_1 and F_2 as independent factors in a 2^2 full factorial and generating the third factor via a defining relation, such as $F_3 = F_1F_2$. This design is shown in Table 2. Key observations from this design are: The 4-run design is a $\frac{1}{2}$ fraction of the full 2^3 factorial. The defining relation $F_3 = F_1F_2$ creates confounding (aliasing). The columns of the potential model matrix $\mathbf{A}$ are not all independent: $F_3 \equiv F_1F_2,\ F_1 \equiv F_2F_3,\ F_2 \equiv F_1F_3,\ \mathbf{1}_4 \equiv F_1F_2F_3$. Consequently, the matrix $\mathbf{A}^\top\mathbf{A}$ for the full model is singular. We cannot estimate all eight original parameters separately. To fit a model, we must select a subset of four linearly independent columns $\mathbf{C} \subset \mathbf{A}$ (e.g., $\mathbf{1}_4, F_1, F_2, F_3$) to obtain a nonsingular $\mathbf{C}^\top\mathbf{C}$. The price is that the effect of an interaction is confounded with (indistinguishable from) a main effect. For instance, any estimated effect for factor F_3 actually represents the combined effect of F_3 and the F_1F_2 interaction.*

Table 2: A regular 2^{3-1} FrFD with defining relation $F_3 = F_1F_2$.

	Design Matrix $\mathbf{d}$				*Aliased Model Columns*				
Run	F_1	F_2	F_3	Y	F_1F_2	F_1F_3	F_2F_3	$F_1F_2F_3$	$\mathbf{1}_4$
r_1	-1	-1	$+1$	y_1	$+1$	-1	-1	$+1$	$+1$
r_2	-1	$+1$	-1	y_2	-1	$+1$	-1	$+1$	$+1$
r_3	$+1$	-1	-1	y_3	-1	-1	$+1$	$+1$	$+1$
r_4	$+1$	$+1$	$+1$	y_4	$+1$	$+1$	$+1$	$+1$	$+1$
Aliasing Pattern					$F_1F_2 = F_3$	$F_1F_3 = F_2$	$F_2F_3 = F_1$	$F_1F_2F_3 = \mathbf{1}_4$	

2.3 Optimal Design Selection and Criteria

For any experiment with n runs and s_k q_k-level factors, $1 \leq k \leq m$, the corresponding experimental design is an $n \times s(= s_1 + s_2 + ... + s_m)$ matrix such that the s_k representative columns of the q_k-level factors take values from the set $\{0, 1, 2..., q_k - 1\},\ k = 1, 2, ..., m$. The set of all these possible experimental designs is denoted as $\mathbb{D}_n\left(q_1^{s_1}q_2^{s_2}...q_m^{s_m}\right)$. It is obvious that the set $\mathbb{D}_n\left(q_1^{s_1}q_2^{s_2}...q_m^{s_m}\right)$ contains a huge number of possible experimental designs and thus the selection of optimal experimental designs from it, especially for large $n,\ s_k,\ q_k$ is an NP-hard optimization problem. The significant problem experimenters may face is the construction of optimal FrFDs from the set of all the possible FrFDs, which reduce the experimental cost and provide more efficient information about the behavior of the phenomena under the experimentation. From the practical point of view, constructing such optimal FrFDs is the most important and difficult part for investigators, especially for experiments with large numbers of factors, runs and levels.

Example 2.3 *For a simple example, when $n = 3,\ q_1 = 3,\ s_1 = 1$ and $s_k = 0,\ k = 2, ..., m$ the searching process for an optimal design with three runs and a factor with three levels $\boldsymbol{d}^\star \in \mathbb{D}_3\left(3^1\right)$ will be over the following possible 27 columns (designs with one factor)*

$$\mathbb{D}_3\left(3^1\right) = \left\{\begin{array}{cccccccccccccccccccc|cccccc} 0 & 1 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 0 & 0 & 1 & 1 & 2 & 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 1 & 2 & 1 \end{array}\right\}. \tag{4}$$

Searching for an optimal FrFD in $\mathbb{D}_n\left(q_1^{s_1}q_2^{s_2}...q_m^{s_m}\right)$ needs to compare $(q_1)^{ns_1}(q_2)^{ns_2}\ ...\ (q_m)^{ns_m}$ FrFDs to find the best one which optimize (from a given perspective) a given criterion. Thus, searching for an optimal FrFD is a significant hard problem in the sense of computation complexity. Several approaches are adopted to reducing the computational complexity. These approaches are not mutually exclusive and often several are used together. The popular two of these approaches are outlined below which are used in this study:

Reducing the Cardinality of the Experimental Domain: For reducing the computation complexity, an efficient distribution of the experimental points has to be given instead of optimizing over the set of all possible points. It is obvious from 4 that, there are some columns that provide no useful information about the different behavior of the system under the experimentation and thus these options can be ignored from the searching domain, such as $\{0\ 0\ 0\}^\top$ which provide information about the behavior of the system under the first level only. Instead of optimizing over the set of all possible experimental points, one may obtain an efficient distribution of the experimental points by considering a much smaller candidate set provided that it contains designs with good performance. The so-called balanced-levels (U-type) [20] is a widely used structure of the experimental points for constructing optimal experimental designs that cover as different as possible behavior of the phenomenon under the experimentation. A U-type experimental designs with n runs and s_k

q_k-level factors, $1 \leq k \leq m$ is an $n \times s (= s_1 + s_2 + ... + s_m)$ matrix such that the entries in each column of the s_k representative columns of the q_k-level factors take values from the set $\{0, 1, 2..., q_k - 1\}$, $k = 1, 2, ..., m$ equally often. The set of all the possible U-type experimental designs is denoted as $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$. Searching for an optimal balanced-levels design $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$ needs to compare $(q_1!)^{\frac{ns_1}{q_1}} (q_2!)^{\frac{ns_2}{q_2}} ... (q_m!)^{\frac{ns_m}{q_m}}$ balanced-levels designs. For the simple example in (4), the searching process for an optimal balanced-levels design with three runs and a factor with three levels $\mathbb{U}_3\left(3^1\right)$ will be over the following balanced-levels 6 columns (designs with one factor)

$$\mathbb{U}_3\left(3^1\right) = \left\{ \begin{array}{cccccc} 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 2 & 1 & 2 & 0 & 0 \\ 2 & 0 & 0 & 1 & 2 & 1 \end{array} \right\}. \tag{5}$$

It is worth-mentioning the design from any column in Equation (6) is called a Latin hypercube design (LHD). A LHD [21] is a matrix such that each symbol in any column appears only one time. That is a set LHDs is a subset of the set of U-type designs, i.e., the set of LHDs can be given as follows $\mathbb{U}_n\left(n^s\right)$. Moreover, it is obvious that

$$\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right) \subset \mathbb{D}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right),$$

where

$$\sharp \text{ designs in } \mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} \ ... \ q_m^{s_m}\right) = \prod_{k=1}^{m} (q_k!)^{\frac{ns_k}{q_k}} < \sharp \text{ designs in } \mathbb{D}_n\left(q_1^{s_1} q_2^{s_2} \ ... \ q_m^{s_m}\right) = \prod_{k=1}^{m} (q_k)^{ns_k}.$$

Stopping the Searching Process: Since the optimal (min or max) value of a given optimization criterion is unknown for general cases, finding sharp and tight bounds (lower or upper) of the used criterion for the balanced-levels designs in $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} \ ... \ q_m^{s_m}\right)$ is a significant approach for reducing the computational complexity of searching for optimal balanced-levels designs. If we need to minimize a given criterion and its value of any balanced-level design is equal to its lower bound, then we stop the searching process and the corresponding balanced-levels design is an optimal balanced-levels design with respect to this criterion. However, in some cases the lower bound (LB) cannot be reached. In such circumstances, the efficiency of any balanced design $\mathbf{d} \in \mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$ via a given optimization criterion (Cri) is defined by

$$\mathcal{EFF}(\mathbf{d}) = \frac{LB}{Cri(\mathbf{d})} \leq 1. \tag{6}$$

When the efficiency $\mathcal{EFF} = 1$, the balanced-levels design over the set $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$ is an optimal design in terms of the used optimization criterion, however when the efficiency $\mathcal{EFF}$ is close to 1, the balanced-levels design over the set $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$ is called a nearly optimal balanced-levels design, i.e., good approximation to optimal. Therefore, when $\mathcal{EFF}$ is close to 1, the balanced-levels design is more optimal design. It is obvious that we need to use a tight and sharp lower bound in the searching process. In this context, a foundational contribution was made by [22], who derived a lower bound for the CDisc (cf. Table 3) in regular two-level FrFDs. This work initiated a sustained line of research. [23] subsequently extended these results to non-regular two-level designs, providing lower bounds for both the CDisc and WDisc (cf. Table 3). Further refinements emerged through the work of [24, 25, 26], who established tighter lower bounds for these discrepancy measures. The scope was later broadened to multi-level designs, with [27] and [28] deriving lower bounds for the CDisc for three- and four-level designs, respectively.

Basic Optimization Perspectives and Criteria: Now comes to mind the following logical question: *How to select such optimal U-type designs from the set* $\mathbb{U}_n\left(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}\right)$? From various perspectives, this question has been answered and several kinds of optimal U-type designs with good statistical properties are investigated from the following widely used four basic optimization perspectives:

- *First Perspective: Minimizing the variance of the parameter estimates:* Based on the forgoing discussions for the regression model, we can get the following logical optimization perspective. From Equation(3), we get

$$E\left(\hat{\beta}\right) = \beta \ \text{ and } \ Var(\hat{\beta}) = (\mathbf{A}^\top \mathbf{A})^{-1} \sigma^2. \tag{7}$$

  From Equation 7, we need to minimize the generalized (average) variance of the parameter estimates for a pre-specified model *to get as good as possible parameter estimation "unbiased with minimum variance"*, such as D-optimal designs. Unlike standard classical designs such as FuFDs nd FrFDs, D-optimal designs are always an option regardless of the type of model the experimenter wishes to fit. D-optimal designs are straight optimizations based on a chosen optimality criterion and the model that will be fit. The optimality criterion used in generating D-optimal designs is one of maximizing $|\mathbf{A}^\top \mathbf{A}|$, the determinant of the information matrix $\mathbf{A}^\top \mathbf{A}$. *It is worth mentioning that from the same perspective many criteria have been given, such as A-, E- and G-optimal designs [29]. Moreover, this perspective is a model dependent and thus the experimenter must specify the underlying model before designing the experiment. Therefore, it is still hard to design A-, E-, G-, D-optimal experiments if there is no information about the underlying model or the information is incorrect.*

- *Second Perspective: Minimizing the similarities among the experimental runs:* Minimizing the similarities (i.e., maximizing the dissimilarities or distances) among the experimental runs (points) *to get as different information as possible about the behavior of the experiment, i.e., maximize the space coverage by the design points.* The *Hamming distance (HD)* of two vectors is the number of coordinates in which they differ. Therefore, HD is a good criterion to measure the dissimilarity among the experimental runs for comparing two designs with the same size. For any FrFD with n runs, q levels and s factors $\mathbf{d} \in \mathbb{U}_n(q^s)$, its HD pattern (HDP) is the vector of the n^2 HDs between any two runs that is defined as follows

$$HDP(\mathbf{d}) = (\mathcal{H}_0(\mathbf{d}), \ldots, \mathcal{H}_s(\mathbf{d})), \; \mathcal{H}_r(\mathbf{d}) = \frac{1}{n}\sharp\left\{(i,j) : \mathcal{H}_{ij}(\mathbf{d},\mathbf{d}) = r\right\}, \; 0 \le r \le s, \tag{8}$$

where $\sharp\{.\}$ denotes the cardinality of a set and $\mathcal{H}_{ij}(\mathbf{X}_1, \mathbf{X}_2)$ is the HD between the i^{th} row of the matrix $\mathbf{X}_1$ and the j^{th} row of the matrix $\mathbf{X}_2$. From (8), it is obvious that $\sum_{r=0}^{s} \mathcal{H}_r(\mathbf{d}) = n$. A good design from this point of view is called a minimum HD FrFD that minimizes (sequentially) the HDP over the experimental domain and thus maximizes the dissimilarity among its runs to get as different as possible information about the behavior of the system under the experimentation. *For example*, a design $\mathbf{d}_1 \in \mathbb{U}_{12}\left(q^4\right)$ with $HDP(\mathbf{d}_1) = (0,0,0,7,5)$ is better than a design $\mathbf{d}_2 \in \mathbb{U}_{12}\left(q^4\right)$ with $HDP(\mathbf{d}_2) = (0,0,0,10,2)$. Where the design $\mathbf{d}_1$ has 7 pairs of runs with $HD = 3$ and 5 pairs of runs with $HD = 4$, however the design $\mathbf{d}_2$ has 10 pairs of runs with $HD = 3$ and 2 pairs of runs with $HD = 4$. Therefore, the design $\mathbf{d}_1$ has more pairs with large HD value. Moreover, a maximin distance design (MDD) [8] maximizes the distances between the experimental points by maximizing the minimum L_2-distance (ML_2D) so that no two design points are too close. For any $\mathbf{d} \in \mathbb{U}_n(q^s)$ with n rows $r_i = (r_{i1}, ..., r_{is}),\; 1 \le i \le n$, its ML_2D is defined as follows

$$ML_2D(\mathbf{d}) = \min\left\{L_2D_{ij} = \sum_{k=1}^{s} |r_{ik} - r_{jk}|^2 : \; r_i \ne r_j, \; r_i, r_j \in \mathbf{d}\right\}.$$

A MDD maximizes the ML_2D value among all the ML_2D values over the experimental domain. MDDs represent a prominent class of space-filling designs, known to be asymptotically D-optimal under the Gaussian process model as inter-point correlations diminish[8]. However, their construction poses significant practical challenges. While algorithmic searches, such as those by [30] and [31], offer flexibility in the numbers of runs and factors, they often become computationally prohibitive for large designs. Consequently, recent research has focused on more structured, albeit technical, construction methods [32, 33].

- *Third Perspective: Minimizing the confounding among the experimental factors:* For minimizing the confounding among the experimental factors, the MacWilliams transformation of the HDP that is called the *(generalized) word-length pattern (GWLP)*. For any FrFD with n runs, q levels and s factors $\mathbf{d} \in \mathbb{U}_n(q^s)$, the GWLP is given as follows

$$GWLP(\mathbf{d}) = (\mathcal{A}_1(\mathbf{d}), \ldots, \mathcal{A}_s(\mathbf{d})), \; \mathcal{A}_r(\mathbf{d}) = \frac{1}{n}\sum_{t=0}^{s}\left(\sum_{a=0}^{r}(-1)^a\binom{t}{a}\binom{s-t}{r-a}(q-1)^{r-a}\right)\mathcal{H}_t(\mathbf{d}). \tag{9}$$

It is worth-mentioning that $\mathcal{A}_r(\mathbf{d})$ can be defined in view of the ANOVA model $Y = X_0\beta_0 + X_1\beta_1 + ... + X_s\beta_s + \epsilon$ as follows:

$$\mathcal{A}_r(\mathbf{d}) = \frac{1}{n^2}\sum_{k}\left|\sum_{i}^{n} x_{ik}^{(r)}\right|^2, \; X_r = \left(x_{ik}^{(r)}\right), \; r = 0, 1, ..., s, \tag{10}$$

where Y is the vector of the n observations, β_0 is the intercept, X_0 is an $n \times 1$ all-one vector, β_r is the vector of all r-factor interactions, X_r is the matrix of orthogonal contrast coefficients for β_r and ϵ is the random error. *A minimum aberration design (MAD)* is a widely used class of optimal designs that sequentially minimizes the GWLP among all the designs in the domain [34, 35, 36, 37] for minimizing the confounding among its factor-effects. It is obvious that $\mathcal{A}_0(\mathbf{d}) = 1$ for any design and $\mathcal{A}_1(\mathbf{d}) = 0$ for a U-type design. When $\mathcal{A}_1(\mathbf{d}) = \mathcal{A}_2(\mathbf{d}) = 0$, the design is called an *orthogonal design*. When $\mathcal{A}_1(\mathbf{d}) = \mathcal{A}_2(\mathbf{d}) = ... = \mathcal{A}_t(\mathbf{d}) = 0$, the design is called an *orthogonal array of strength t* [38]. It is obvious that, an orthogonal array of strength t_1 is better than an orthogonal array of length t_2, if $t_1 > t_2$. Moreover, the FuFD with s factors is an orthogonal array of strength s. Moreover, low correlation between columns is another criterion to measure the goodness of a design. Column-orthogonality can ensure that the estimates of the linear effects in a regression model can be uncorrelated [39, 40]. A design is called orthogonal if the correlation coefficient between any two distinct columns in the design is zero. For any design $\mathbf{d} \in \mathbb{U}_n(q^s)$ with s columns $c_j = (c_{1j}, ..., c_{nj}),\; 1 \le j \le s$ and levels $\{..., -1, 0, 1, ...\}$ for odd q and $\{..., -1, 1, ...\}$ for even q, its orthogonality is defined as follows

$$\rho_{max}(\mathbf{d}) = \max\left\{\rho_{ij}(\mathbf{d}) = \rho(c_i, c_j) : \; c_i \ne c_j\right\}, \; \rho(c_i, c_j) = \left|\frac{\sum_{k=1}^{n}(c_{ki} - \overline{c_i})(c_{kj} - \overline{c_j})}{\sqrt{\sum_{k=1}^{n}(c_{ki} - \overline{c_i})^2 \sum_{k=1}^{n}(c_{kj} - \overline{c_j})^2}}\right|, \; \overline{c_i} = \sum_{k=1}^{n}\frac{c_{ki}}{n}.$$

It is worth mentioning that from the same perspective many criteria have been given, such as the *Resolution* for regular FrFD that is the length of the shortest relation, i.e., the number of factors in the relation. The O-criterion pattern (OP)[23], NB-criterion pattern (NBP)[41], the deviation criterion pattern (DP)[42] and the χ^2-criterion pattern (χ^2P)[43] are another criteria to assess the non-orthogonality among any r factors.

- *Fourth Perspective: Filling the experimental domain with as few gaps as possible:* Filling the experimental domain with as few gaps as possible *to understand all the possible different behaviors of the experiment by covering all of its domain*, such as *space-filling designs*, e.g., uniform designs (UDs). UDs [10, 44] are an optimal class of space-filling designs that are widely used in several real-life applications, such as dynamic systems [45], chemistry and chemical engineering [46], and computer experiments [47], chemometrics [48]. These designs scatter their points equally on the experimental domain by minimizing the deviation (called, discrepancy) between the empirical distribution of the selected points and the theoretical uniform distribution. To do that, let $F(\mathbf{x})$ be the distribution function of the uniform distribution on the unit cube $C^s = [0,1]^s$ and $F_n(\mathbf{x})$ be the empirical distribution of a set $\mathbf{d}$ of n s-dimensional points on $C^s = [0,1]^s$. $\mathbf{d}$ is called a UD, if it minimizes the following star-discrepancy [49] over the experimental domain

$$Disc^{\star}(\mathbf{d}) = \max_{\mathbf{x}\in C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|. \tag{11}$$

The L_p-star-discrepancy is a significant extension of the star-discrepancy that is defined as follows

$$Disc_p^{\star}(\mathbf{d}) = \left(\int_{C^s} |F_n(\mathbf{x}) - F(\mathbf{x})|^p\right)^{\frac{1}{p}}.$$

When $p = 2$, the computational complexity of $Disc_2^{\star}(\mathbf{d})$ is reasonable and [50] gave an analytic formula for it. [51, 52] employed the so-called reproducing kernel of Hilbert space for generating various of discrepancies. Let $\mathfrak{X}$ be an experimental domain, which is a measurable subset of R^s and $\mathcal{K}(\mathbf{x},\mathbf{y})$ be a kernel function defined on $\mathfrak{X}\times\mathfrak{X}$ satisfying $\mathcal{K}(\mathbf{x},\mathbf{y}) = \mathcal{K}(\mathbf{y},\mathbf{x}),\ \forall \mathbf{x},\mathbf{y}\in\mathfrak{X}$ and $\sum_{i,j=1}^{s} bib_j\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) \geq 0,\ \forall \mathbf{x}_i,\mathbf{x}_j \in \mathfrak{X}$ and $b_i, b_j \in R$. For a design with n runs (points) $\mathbf{d} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ over the domain $\mathfrak{X}$, the L_2-type discrepancy for a given kernel is defined as

$$Disc_2^{\star}(\mathbf{d}) = \sqrt{\int_{\mathfrak{X}\times\mathfrak{X}} \mathcal{K}(\mathbf{x},\mathbf{y})dF_{\mathbf{u}}(\mathbf{x})dF_{\mathbf{u}}(\mathbf{y}) - \frac{2}{n}\sum_{i=1}^{n}\int_{\mathfrak{X}} \mathcal{K}(\mathbf{x}_i,\mathbf{y})dF_{\mathbf{u}}(\mathbf{y}) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)},$$

where $F_{\mathbf{u}}(.)$ is the uniform distribution in the experimental domain $\mathfrak{X}$. Commonly used reproducing kernels for discrepancies in the literature are defined on $\mathfrak{X} = [0,1]^s$ and have a multiplicative form $\mathcal{K}(\mathbf{x},\mathbf{y}) = \prod_{k=1}^{s} f(x_k, y_k)$, where $f(x,y)$ is defined on $[0,1]^2$. Then, the corresponding discrepancy can be expressed by

$$Disc_2^{\star}(\mathbf{d}) = \sqrt{\Psi - \frac{2}{n}\sum_{i=1}^{n}\prod_{k=1}^{s} f(x_{ik}) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\prod_{k=1}^{s} f(x_{ik}, x_{jk})}, \tag{12}$$

where $f(x) = \int_{y=0}^{1} f(x,y)dy$ and $\Psi = \int_{\mathfrak{X}\times\mathfrak{X}} \mathcal{K}(\mathbf{x},\mathbf{y})dF_{\mathbf{u}}(\mathbf{x})dF_{\mathbf{u}}(\mathbf{y})$ is a constant. Various discrepancies are given for generating UDs by selecting different kernel functions. The wrap-around L_2-discrepancy (WDisc) [51, 52], centered L_2-discrepancy (CDisc) [51, 52], discrete discrepancy (DDisc)[53], Lee discrepancy (LDisc) [54], and mixture L_2-discrepancy (MDisc) [55] are all the widely used discrepancies in practice . Table 3 gives the kernel functions of the mentioned discrepancies. It is worth mentioning that, the DDisc only considers the dissimilarity between two runs and does not measure how far between different values of two runs, i.e., DDisc can be used for experiments with nominal factors and does not suitable for multi-level experiments. However, the LDisc is proposed as an extension of the DDisc for measuring how far between the values of two runs in multi-level and mixed-level designs. A comprehensive review about the selection of these discrepancies can be found in [56].

Table 3: Kernel functions for various discrepancies

Discrepancy	$f(x,y)$	Types of factors
$DDisc(a,b)$	$\begin{cases} a, & \text{if } x = y, \\ b, & \text{if } x \neq y,\ a > b > 0. \end{cases}$	Qualitative factors
LDisc	$1 - \min\{\lvert x-y\rvert, 1-\lvert x-y\rvert\}$	Qualitative factors
WDisc	$\frac{3}{2} - \lvert x-y\rvert(1-\lvert x-y\rvert)$	Quantitative factors
CDisc	$1 + \frac{1}{2}\lvert x-\frac{1}{2}\rvert + \frac{1}{2}\lvert y-\frac{1}{2}\rvert - \frac{1}{2}\lvert x-y\rvert$	Quantitative factors
MDisc	$\frac{15}{8} - \frac{1}{4}\lvert x-\frac{1}{2}\rvert - \frac{1}{4}\lvert y-\frac{1}{2}\rvert - \frac{3}{4}\lvert x-y\rvert + \frac{1}{2}\lvert x-y\rvert^2$	Quantitative factors

3 Algorithmic Search Techniques for Constructing Optimal Designs

The task of obtaining an efficient experimental design based on any of the above-mentioned four optimization perspectives is a major challenge in scientific research. For example, an optimal design from the uniformity perspective is called a uniform design that is the design $\mathbf{d}^* \in \mathbb{U}_n(q^s)$ that minimizes the objective function (Disc) on $\mathbb{U}_n(q^s)$, i.e.,

$$Disc(\mathbf{d}^*) = \min_{\mathbf{d}\in\mathbb{U}_n(q^s)} Disc(\mathbf{d}). \tag{13}$$

There are many useful stochastic optimization algorithms that can help us to find a solution for Equation 13. The simulated annealing (SA) algorithm [57] is the first one. However, most of uniform designs have been obtained by the threshold accepting (TA) algorithm. TA algorithm is a simplified version of the SA. [58] first applied TA as a powerful tool for UD construction. More comprehensive discussion can be found in [45]. A large number of UDs have been obtained via TA and listed on the website: `http://uic.edu.hk/isci/UniformDesign`. These designs have been widely used in industries.

3.1 Threshold Accepting (TA) Algorithm

TA algorithm starts with an initial design $\mathbf{d}_{initial} \in \mathbb{U}_n(q^s)$ which might be randomly chosen on $\mathbb{U}_n(q^s)$. Then, a large number of iterations are performed. In each iteration, TA algorithm evaluates whether to replace its current design $\mathbf{d}_{current} \in \mathbb{U}_n(q^s)$ by a new $\mathbf{d}_{new} \in \mathbb{U}_n(q^s)$, which is randomly chosen from a small perturbation of $\mathbf{d}_{current}$. In other words, $\mathbf{d}_{new}$ is randomly taken in the neighborhood of $\mathbf{d}_{current}$. Let $Disc(.)$ be the objective discrepancy function, the replacement of $\mathbf{d}_{current}$ by $\mathbf{d}_{new}$ is taken in the i^{th} stage if

$$\Delta Disc = Disc(\mathbf{d}_{new}) - Disc(\mathbf{d}_{current}) \leq T_i, \tag{14}$$

where T_i is a given threshold value in the threshold sequence $T = [T_1 \dots T_I]$. The elements in T are non-negative real numbers and decreasing to 0 during the process, i.e. $T_1 > \dots > T_I = 0$. This setting allows $\mathbf{d}_{current}$ to reach and left bad local optimums while ending up in a local optimum with good quality. Under each stage, a number of J iterations have to be performed so that the system goes into a stable state. Two positive integers I and J, in this process are pre-decided. Finally after IJ iterations, TA outputs the last $\mathbf{d}_{current}$ as final optimal design, denoted by $\mathbf{d}_{optimal}$.

Although TA algorithm is well acknowledged for UD construction, certain flaws of this algorithm should not be neglected. Specifically, the TA algorithm sometimes fails to obtain a $\mathbf{d}_{optimal}$ better than $\mathbf{d}_{initial}$, i.e., $Disc(\mathbf{d}_{initial}) < Disc(\mathbf{d}_{optimal})$. Meanwhile, some elements in the threshold sequence (T) are redundant if the setting I is large. These weaknesses inhibit the performance of the TA algorithm especially for searching UD with large size. For example, TA algorithm failed to provide a UD on the domain $\mathbb{U}_{27}(3^{13})$.

3.2 Adjusted TA Algorithm

[59] introduced the following problems and solutions for the TA Algorithm. The details about the searching mechanism of the TA algorithm have been introduced above. Although the frame of this algorithm is regulated, several settings may be different from case to case. We briefly introduce the settings of the TA algorithm, then discuss the influence of initial design ($\mathbf{d}_{initial}$) and the problems encountered by this algorithm. To utilize TA, we should input the objective function (say, discrepancy function $Disc(.)$), the initial design ($\mathbf{d}_{initial}$), the number of thresholds (I), the number of iterations under one threshold (J), as well as determine (i) the searching domain, (ii) the neighborhood for $\mathbf{d}_{current}$ and (iii) the threshold sequences. The details for the above three issues are addressed in [59] as follows:

- *Searching domain:* The searching domain of the TA algorithm is a set of designs $\mathbb{U}_n(q^s)$. Each element of this set is a U-type design with n runs and s factors each having q levels. It is a $n \times s$ matrix each column taking values from $\{0, 1, 2, \dots, q-1\}$ equally often.
- *Neighborhood:* The neighborhood of $\mathbf{d}_{current}$ is defined as follows

$$N(\mathbf{d}_{current}) = \{\mathbf{d} : \mathbf{d} \text{ is a design by exchanging two elements in a randomly picked column of } \mathbf{d}_{current}\}.$$

- *Threshold sequence:* To form a threshold sequence $T = [T_1, \dots, T_I]$ in the TA algorithm, a set of M designs are randomly generated. Let R be the range of the objective function values (Disc-values) of these M designs. The first threshold (T_1) is chosen as a fraction $0 < \alpha < 1$ of R and the remaining $I - 1$ thresholds diminish with a given procedure (linear or nonlinear) towards zero. One can choose several α values in advance and find the best one.

How initial design ($\mathbf{d}_{initial}$) affects the quality of the output design ($\mathbf{d}_{optimal}$) is an important issue in the TA algorithm. There are 2 possible strategies to choose $\mathbf{d}_{initial}$ as given in [59].

- *Strategy 1[59]:* We randomly choose m initial designs and run the TA algorithm k times. The best output among the m output designs is recommended as $\mathbf{d}_{initial}$.
- *Strategy 2[59]:* We utilize the TA algorithm with several phases. In each phase, the settings of the TA algorithm remain the same except for the scale parameter α that affects the threshold sequence. That is, in the first phase, traditional the TA algorithm is used with $\alpha = \alpha_1$. The output design is recorded as $\mathbf{d}_1$. In the second phase, we set this $\mathbf{d}_1$. as the new initial design and utilize the TA algorithm again with $\alpha = \alpha_2$. A new output, $\mathbf{d}_2$, can be obtained. Then use $\mathbf{d}_2$ as the initial design for the third phase with the same procedure. So on and so forth, we stop the process and obtain the final optimal design $\mathbf{d}_k$ at the end of the k^{th} phase when $|Disc(\mathbf{d}_k) - Disc(\mathbf{d}_{k-1})|$ is very small. According to the experience, $\alpha_1, ..., \alpha_k$ should set as a descending array to ensure the convergency of this algorithm.

The rationale behind strategy 2 is to provide TA a $\mathbf{d}_{initial}$ in a good condition. With a good start point, the TA algorithm is more likely to provide abetter $\mathbf{d}_{optimal}$ if we choose α accordingly. The following example compares performance of the two aforementioned strategies.

Example 3.1 *[59] generated a random sample in $\mathbb{U}_{27}(3^{13})$ as an initial design $\mathbf{d}_{initial}$, then utilize the TA algorithm with strategies 1 and 2. The setting of strategy 1 is $(I = 20,\ J = 5000,\ \alpha = 0.15, m = 5)$; and the setting of strategy 2 is $(I = 20,\ J = 5000,\ \alpha = [\alpha_1, ..., \alpha_5] = [0.15\ 0.016\ 0.01\ 0.002\ 0.0005])$. For each strategy, [59] independently run the TA algorithm 30 times with objective function MDisc. The best UD in strategy 2 $(MDisc(\mathbf{d}_{2optimal}) = 64.3689)$ has a smaller MDisc value than the best UD in strategy 1 $(MDisc(\mathbf{d}_{1optimal}) = 64.4311)$. Meanwhile, the median MDisc design from strategy 2 $(MDisc(\mathbf{d}_{2median}) = 64.4523)$ is better than the median MDisc from strategy 1 $(MDisc(\mathbf{d}_{1median}) = 64.5911)$ as well. Therefore, strategy 2 is recommended to construct UD. Figure 1 provides a MDisc trace from the first attempt by strategy 2. This plot shows the objective function value in each iteration step and is useful to explore the behavior of iteration process. The traces of 5 phases are plotted in different colors. The optimal design in phase i is recorded as $\mathbf{d}_i, i = 1, ..., 5$. It is clear from the plot $\mathbf{d}_5$ $(MDisc(\mathbf{d}_5) = 64.4338)$ improves $\mathbf{d}_{initial}$ $(MDisc(\mathbf{d}_{initial}) = 64.8721)$ significantly. However, we can still observe two problems as given in [59].*

- *Problem 1[59]. When $\mathbf{d}_{initial}$ is a design with low Disc-value, the TA algorithm may give a $\mathbf{d}_{optimal}$ with a larger Disc-value than $\mathbf{d}_{initial}$. From phase 3 in Figure 1, [59] found the optimal design $(MDisc(\mathbf{d}_3) = 64.5031)$ fails to improve its initial design $(MDisc(\mathbf{d}_2) = 64.3746)$. This problem happens because TA always encourages the current design ($\mathbf{d}_{current}$) to jump out from a local optimum if the current threshold value (T) is not small. This property is helpful for an inferior $\mathbf{d}_{initial}$. However, if $\mathbf{d}_{initial}$ is in a good quality, with a large T, the TA algorithm inclines to abandon the vicinity of $\mathbf{d}_{initial}$ and searches designs in other regions. It is possible that no $\mathbf{d}_{current}$ in the subsequent search is better than $\mathbf{d}_{initial}$.*
- *Problem 2[59]. The last several thresholds are lack of efficiency when I is large. Traditionally, an exponential cooling schedule is preferred for TA. Once the first threshold, say T_1, establishes, the following threshold values are pre-decided as*
$$T_i = \frac{I-i}{I} T_{i-1},\ i = 2, ..., I.$$
In Figure 1 phase 1, threshold values are recorded as $T_1 = [T_{1,1}, ..., T_{1,20}]$. Note that the last several thresholds $T_{1,16} = 4 \times 10^{-5},\ T_{1,17} = 5 \times 10^{-6},\ T_{1,18} = 5 \times 10^{-7}$ and $T_{1,19} = 2 \times 10^{-8}$ that are too close to zero. These infinitesimal threshold values may hamper the searching of TA. In trace plot, it is conspicuous that the TA algorithm is inhibited during the last several thresholds where it can merely make any improvements. Because it is a necessity to use large I in the TA algorithm when optimizing large designs, the problem occurs here is inevitable.

[59] presented the adjusted TA (ATA) algorithm in order to solve the above problems as follows:

- *Solution of Problem 1[59].* To deal with Problem 1, [59] proposed a new mechanism called historical optimum reversion (HOP). This mechanism allows the current design ($\mathbf{d}_{current}$) to return to the historical optimal design of the TA algorithm at certain moments. More specifically, [59] embed a "judgement " before the iterations under each threshold values. That is, once the threshold value changes in the TA algorithm, a comparison will be made immediately on $Disc(\mathbf{d}_{current})$ and $Disc(\mathbf{d}_{historical})$, where $\mathbf{d}_{historical}$ is the historical optimal design of the TA algorithm. If $Disc(\mathbf{d}_{historical}) > Disc(\mathbf{d}_{current})$, we will let $\mathbf{d}_{current} = \mathbf{d}_{historical}$ and proceed the ensuing iterations.
- *Solution of Problem 2[59].* For designs with a relatively large I, the last several thresholds are too close to zero and inefficient. Thus, we propose a mixture thresholds schedule to fix this problem. This schedule allows the

exponential schedule to transfer to a linear schedule when the threshold value is too small. Consider c $(0 < c < 1)$ as a parameter for schedule switching. In the mixture schedule, when $T_i > cT_1 (i = 2, ..., I)$, the threshold values in the TA algorithm reduce in an exponential manner. As the first $T_i \leq cT_1$ $(i = 2, ..., I)$ occurs, the ensuing threshold values will forsake the exponential manner and diminish linearly towards zero as

$$T_i = \frac{I - i - 1}{I - i} T_{i-1},\ i = 2, ..., I,\ T_{i-1} \leq cT_1.$$

The following example (source [59]) gives an illustration to the effectiveness of the 2 adjustments:

Example 3.2 *[59] tried to optimize $\mathbf{d}_{initial}$ (generated from Example 3.1) again with the ATA algorithm. The settings are the same as in Example3.1 strategy 2. Set $c = 0.03$. From only one trial, we obtain $\mathbf{d}^*_{2optimal}$ with $MDisc(\mathbf{d}^*_{2optimal}) = 64.1888$. This result already outperforms the best attempt in Example3.1 $MDisc(\mathbf{d}_{2optimal}) = 64.3689)$. Thus we can conclude the ATA algorithm has some advantages. The trace plot for this trial is provided in Figure 2. In phases 2 and 3 of this example, [59] observed that HOP eliminates the variation of the TA algorithm. Although $\mathbf{d}_{current}$ evacuates from $\mathbf{d}_{initial}$ like before, this mechanism allows the TA algorithm to retrieve its historical optimum from time to time. It also guarantees $Disc(\mathbf{d}_{optimal}) = Disc(\mathbf{d}_{initial})$. Therefore Problem 1 has been solved. For solving Problem 2, [59] focused on the trace plot (Figure 2). As [59] pointed out, the iterations under the last several thresholds are inhibited by the exponential threshold schedule. Under the mixture threshold schedule, TA is more active for these iterations. In Figure 2 phase 1, [59] detected a significant improvement during iterations under $I = 15$ (the 75000^{th} to 79999^{th} iterations). It is obviously a consequence of employing the mixture threshold schedule since this phenomenon cannot occur under a threshold close to zero.*

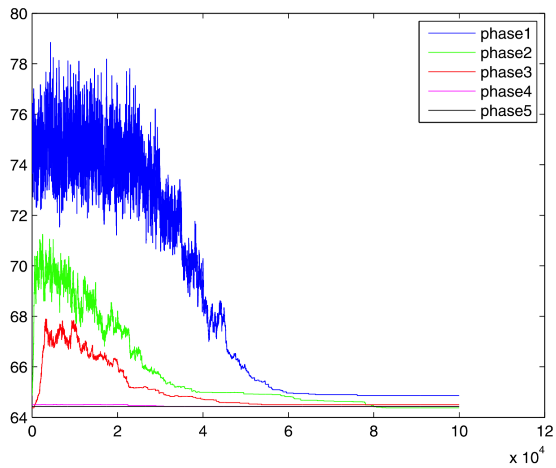


Figure 1: The MDisc trace plot of one trial of the classical TA algorithm in Strategy 2 of Example 3.1. Source [59]

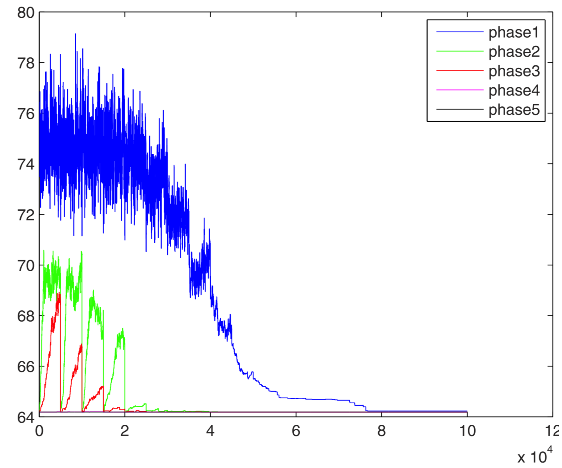


Figure 2: The MDisc trace plot of the new ATA algorithm by phase in Example 3.2. Source [59]

Since advantages of ATA algorithm have been revealed, [59] used it to update the website UD tables by designs with better uniformity. Furthermore, the ATA algorithm helps us to obtain a new UD with 27 runs and 13 factors each having 3 levels under the MDisc. This UD is a minimum aberration design. For example, Table 4 (source [59]) gives the MDisc values of the designs that are generated via the new ATA algorithm and the classical TA algorithm. The results indicate that the new ATA algorithm is much better than the classical TA algorithm.

Table 4: Comparison of MDisc for designs obtained from the new ATA algorithm and the traditional TA algorithm. Source [59]

$\mathbf{d} \in \mathbb{U}_n(n^{n-1})$	$n = 24$	$n = 25$	$n = 26$	$n = 27$	$n = 28$
MDisc via the new ATA algorithm	9.8743×10^3	1.7212×10^4	2.9833×10^4	5.1878×10^4	8.9756×10^4
MDisc via the classical TA algorithm	9.8765×10^3	1.7213×10^4	2.9838×10^4	5.1880×10^4	8.9768×10^4
Difference	2.19	1.43	5.56	2.47	11.53
$\mathbf{d} \in \mathbb{U}_n(n^{n-1})$	$n = 29$	$n = 30$	$n = 31$	$n = 32$	$n = 33$
MDisc via the new ATA algorithm	1.5579×10^5	2.6913×10^5	4.6666×10^5	8.0513×10^5	1.3938×10^6
MDisc via the classical TA algorithm	1.5581×10^5	2.6914×10^5	4.6676×10^5	8.0517×10^5	1.3940×10^6
Difference	17.81	16.64	101.60	41.04	228.18

3.3 Efficient Lower Bounds as Benchmarks for Stopping the Searching Process

As evident from Equation 6, the lower bounds of the discrepancies can serve as benchmarks to terminate the search process. If a design achieves its lower bound or comes close to it, it qualifies as a uniform or nearly uniform design with respect to that discrepancy. The following discussion summarizes key recent results concerning efficient lower bounds for various discrepancies; explicit forms of these bounds are omitted here to save space but can be found in the cited references.

- [60] derived a new lower bound for the CDisc for four-level balanced-level (U-type) designs. This bound is sharper and applicable to a wider range of designs than existing lower bounds in [27] (see Figure 3, source [60]), providing a valuable complement to the existing lower bounds of discrepancies. They also provided a necessary condition for the existence of uniform designs meeting this lower bound.
- [61] conducted an in-depth analysis of the MDisc for symmetric two-, three-, and four-level U-type designs and provided new analytical expressions. Based on these formulations, they presented new lower bounds for the MDisc for symmetric two-, three-, and four-level U-typ designs. They also described necessary conditions for the existence of a uniform design attaining these lower bounds. The results demonstrate that the new lower bounds are more useful and sharper than those given in [55] and [62] for two-level balanced designs, and sharper than those in [62] for three-level U-typ designs (see Figures 4 and 5, source [61]).
- [63] investigated a new analytical expression for the CDisc for mixed two- and three-level U-type designs in detail. Using this formulation, they proposed a new lower bound for the CDisc for U-type designs with mixed two- and three-level factors. They also provided a necessary condition for the existence of uniform designs meeting this lower bound. To illustrate the application of their theoretical results, they tabulated a catalog of lower bounds for U-type designs.
- [64] examined the MDisc as a uniformity measure for asymmetric mixed two- and three-level U-type designs. They provided new analytical expressions based on row distance and a new lower bound for the MDisc for asymmetric level designs. Using this new formulation and lower bound as a benchmark, they implemented an improved version of the fast local search heuristic threshold accepting. This heuristic yields mixed two- and three-level U-type designs with low discrepancy.
- [65] presented new and efficient analytical expressions—in terms of Hamming (row) distance—and lower bounds for the LDisc, WDisc, CDisc, and MDisc for balanced designs involving mixtures of factors with two, three, and four levels. They provided necessary conditions for the existence of these lower bounds. As special cases of these results, they gave many analytical expressions and new lower bounds for symmetric and asymmetric balanced designs. The significance of the new lower bounds was evaluated by comparing their results with the existing literature.
- [66] extended the above work to symmetric designs with q-level factors and asymmetric designs with mixtures of two- and q-level factors, providing new analytical expressions and lower bounds for the WDisc. [67] further extended the work of [66] to asymmetric designs with mixtures of q_1- and q_2-level factors, also giving new analytical expressions and lower bounds for the WDisc. [68] and [69] extended the work of [67] to the most general case involving any number of factors with any number of different levels. They proposed novel analytical expressions and lower bounds for the WDisc and LDisc, and evaluated their significance by comparison with existing literature.

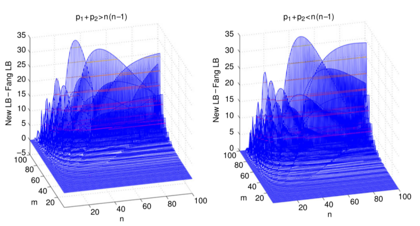


Figure 3: The difference between the new lower bound of CDisc in [60] and the lower bound in [27]. Source [60]

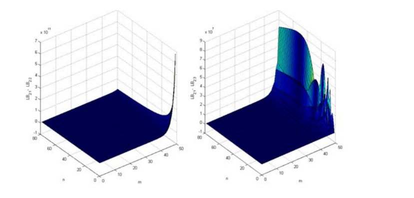


Figure 4: For two-level balanced designs, the lower bound of MDisc given in [61] is more useful and sharper than the lower bounds given in [55] and [62]. Source [61]

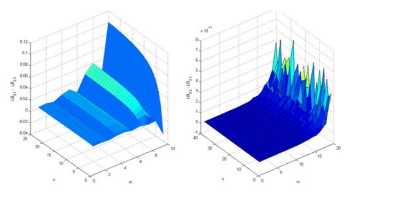


Figure 5: For three-level balanced designs, the lower bound of MDisc given in [61] is more useful and sharper than the lower bounds given in [62]. Source [61]

3.4 Level Permutation and Factor Projection (LPFP) Algorithm

A new method based on level permutations and factor projections (LPFP) is presented to improve the uniformity of a given design in full-dimensional space and to select optimal sub-designs in lower dimensions [70, 71, 72, 73, 74, 75]. The level permutation technique—which randomly permutes the levels in one or more factors to generate a new design—can alter the geometric structure (uniformity) of a design while keeping the GWLP unchanged. Thus, this method can be used to further improve the results obtained from the TA or ATA algorithms.

Let $\mathbf{d}_{\text{optimal}}$ be the output from the TA or ATA algorithms, and let $\mathbb{L}$ and $\mathbb{V}$ denote the sets of all level permutations of $\mathbf{d}_{\text{optimal}}$ and their corresponding discrepancy values, respectively. All permutations in $\mathbb{L}$ can be obtained, and a new optimal design $\mathbf{d}^*_{\text{optimal}}$ can be recorded where

$$\text{Disc}(\mathbf{d}^*_{\text{optimal}}) = \min_{\mathbf{d}\in\mathbb{L}} \text{Disc}(\mathbf{d}). \tag{15}$$

While this technique incurs a substantial computational burden, it would be imprudent to forgo it and thereby miss the opportunity to improve the designs. Moreover, the average of all discrepancy values in $\mathbb{V}$ can serve as a benchmark for selecting the best design. Furthermore, the projections of the new optimal design $\mathbf{d}_{\text{optimal}}$ onto all possible low-dimensional subspaces can be combined with level permutations to select optimal designs with at most s factors.

[76] employed the LPFP algorithm to recommend new uniform minimum aberration and minimum Hamming distance designs with $3 \leq k \leq 13$ factors, 27 runs, and three levels (i.e., $\mathbf{d} \in \mathbb{U}_{27}(3^k)$) for investigating either qualitative or quantitative factors. These designs outperform both existing recommended designs in the literature and existing uniform designs. The authors applied the LPFP algorithm to all 68 non-isomorphic orthogonal designs from $\mathbb{U}_{27}(3^{13})$, which are given by affine resolvable design theory in [77]. Two designs are considered isomorphic if one can be obtained from the other by reordering runs, relabeling factors, or switching the levels of one or more factors.

Tables 5 and 6 (source [76]) provide pseudo-code for obtaining the new recommended designs using the LPFP algorithm and the corresponding computational times, respectively. A comparative study between the new recommended designs by [76] and existing designs is presented. Tables 7 and 8 (from [76]) display the MDisc, CDisc, WDisc, HDP, and GWLP for existing CDisc- and WDisc-based uniform designs (UDs) with three levels, 27 runs, and $3 \leq k \leq 13$ factors from the UD webpage. Table 9 (source [76]) gives these metrics for existing recommended designs in the literature. Tables 10, 11, and 12 (source [76]) provide the corresponding metrics for the new recommended designs based on MDisc, CDisc, and WDisc, respectively. The results demonstrate that the new recommended designs in [76] perform better than existing recommended designs in [59, 70, 73, 78, 79, 80, 81, 82] and the UDs listed on the UD webpage at http://sites.stat.psu.edu/ril4/DMCE/UniformDesign/.

3.5 Bridges among Optimal Designs for Reducing the Computational Search

A significant challenge investigators face is selecting appropriate criteria for constructing efficient experimental designs. As previously mentioned, numerous criteria have been proposed for this purpose. Since some criteria possess greater conceptual simplicity and computational advantages than others, their mutual consistency enables the effective study of difficult problems—such as detecting combinatorial and geometrical isomorphism among designs and selecting optimal designs—by focusing on the simplest criterion. [83] established analytical linkages among these criteria both before and after applying the level permutation algorithm. The paper provides a general framework for investigating connections between criteria, presents conditions for constructing a uniform minimum aberration orthogonal array, and offers recommendations for specific circumstances. Since certain criteria are simpler and computationally more efficient, the study in [83] facilitates in-depth investigation of challenging problems—such as detecting non-isomorphic designs, sorting designs, and selecting optimal designs—by employing the simplest criterion rather than more complex alternatives. The theoretical relationships among various designs are detailed in [83]; their main conclusions are as follows:

- For any design with quantitative factors $\mathbf{d} \in \mathbb{U}_n(q^s),\ q > 2$, level permutations can produce different statistical properties and may improve design quality. However, level permutations do not affect the quality of designs with qualitative factors; that is, all level-permuted versions of a qualitative-factor design yield the same value under any criterion.
- For any design with s quantitative or qualitative factors $\mathbf{d} \in \mathbb{U}_n(q^s),\ q > 2$, reordering factors or runs does not alter the geometric structure or statistical properties of $\mathbf{d}$. Hence, level permutations are unnecessary for designs with qualitative factors, and factor or run permutations are unnecessary for any such design.

For a design $\mathbf{d} \in \mathbb{U}_n(q^s)$ with s quantitative factors, let $\mathbb{L}(\mathbf{d})$ denote the set of all $(q!)^s$ level-permuted designs of $\mathbf{d}$. For a given discrepancy (Disc), the average WDisc (AWDisc), average CDisc (ACDisc), and average MDisc (AMDisc) are defined as:

$$\text{ADisc}(\mathbf{d}) = \frac{1}{(q!)^s} \sum_{\mathbf{d}_l\in\mathbb{L}(\mathbf{d}),\ 1\leq l\leq (q!)^s} [\text{Disc}(\mathbf{d}_l)]^2 \,.$$

For any design $\mathbf{d} \in \mathbb{U}_n(q^s)$, the relationships identified in [83] are summarized in Figure 6 (source [83]):

Table 5: Pseudo-code for selecting UDs using the LPFP algorithm of the 68 non-isomorphic ODs. Source [76]

for $\zeta = 1, \ldots, 68$ **do**
  Generate $\mathbf{d}_\zeta^1, ..., \mathbf{d}_\zeta^\delta$ ▷ Obtain all the $\delta = \binom{13}{k}$ k-factor projected designs
  for $j = 1, \ldots, \delta$ **do**
    Generate $\mathbf{d}_\zeta^{j:1}, ..., \mathbf{d}_\zeta^{j:\rho}$ ▷ Obtain all the $\rho = 3^k$ effective level permuted designs
    $Disc_\zeta^j = \left[Disc(\mathbf{d}_\zeta^{j:1}), \ldots, Disc(\mathbf{d}_\zeta^{j:\rho})\right]'$ ▷ Obtain the discrepancy values for $\mathbf{d}_\zeta^{j:1}, ..., \mathbf{d}_\zeta^{j:\rho}$
    $MinDisc_\zeta^j = min(Disc_\zeta^j)$ ▷ $MinDisc_\zeta^j$ has the smallest value in $Disc_\zeta^j$
  end for
end for
$MinDisc_{all} = \begin{bmatrix} MinDisc_1^1 & \cdots & MinDisc_1^\delta \\ \vdots & \ddots & \vdots \\ MinDisc_{68}^1 & \cdots & MinDisc_{68}^\delta \end{bmatrix}'$
$Disc_{min} = min(MinDisc_{all})$
Find the design $\mathbf{d}_{optimal}$ which satisfies $Disc(\mathbf{d}_{optimal}) = Disc_{min}$
Output $\mathbf{d}_{optimal}$

Table 6: Computational times needed for selecting from $\mathbb{U}_{27}(3^k)$ using the LPFP algorithm in Table 5. Source [76]

♯ factors	♯ projected designs	♯ level permuted designs for each projected design	♯ non-isomorphic subclasses	total time(s)*
13	1	1594323	68	10578
12	13	531441	68	67047
11	78	177147	68	129970
10	286	59049	68	152320
9	715	19683	68	124647
8	1287	6561	68	70161
7	1716	2187	68	29638
6	1716	729	68	9218
5	1287	243	68	2161
4	715	81	68	374
3	286	27	68	44
2	78	9	68	4
1	13	3	68	1

* Programmed by MATLAB (R2009a). CPU: Intel Core (TM) i5-6500 3.20G Hz.
RAM: 32GB. OS: Microsoft Windows 10 64bit.

Table 7: The CDisc, WDisc, MDisc, HDP and GWLP for the existing CDisc UDs on UD webpage. Source [76]

k	CDisc value	MDisc value	WDisc value	HDP	GWLP
3	0.032322467358*	0.108662634793*	0.100143524869*	0 81 162 108*	0 0 0*
4	0.046547389844*	0.234003148857*	0.179335170790*	0 0 162 108 81*	0 0 0 2*
5	0.063525002192	0.474561019546	0.303061546633	0 1 32 168 104 46	0 0 2.7 4.9 0.4
6	0.083475495916*	0.923653472193	0.490921309445	0 0 0 72 189 54 36	0 0 5.3 14 4 2.7
7	0.108697603587	1.760395504466	0.780275746148	0 0 1 13 109 156 60 12	0 0.1 12.2 24.3 21.7 16.5 5.2
8	0.138656965290	3.284945724941	1.216097265937	0 0 0 1 27 159 121 32 11	0 0.3 18.4 49.7 58.5 64.7 42.1 8.3
9	0.175342818719	6.068029034933	1.878057045384	0 0 0 0 6 68 139 115 23 0	0 0.7 30.1 78.9 142.2 200.2 171.1 86.4 18.4
10	0.219131011469	11.062022705536	2.874728849067	0 0 0 0 0 16 96 156 66 16 1	0 1.3 40.9 134.2 284.8 493.6 583.2 419.1 190.8 37.9
11	0.272382988465	20.029155488594	4.358590227002	0 0 0 0 0 2 30 131 135 45 7 1	0 2 56 209.1 524.2 1091.3 1589.1 1552.9 1038.7 421.8 74.9
12	0.336400698080	35.883326976055	6.528325765495	0 0 0 0 0 0 5 47 163 110 23 3 0	0 2.3 75.5 311.6 899.9 2185.2 3810.9 4656.5 4159.9 2527.6 899.2 153.4
13	0.414783285780	64.353103937013	9.843048538280	0 0 0 0 0 0 0 13 83 165 73 15 2 0	0 3.5 96.2 446.1 1477.4 4056.1 8219.2 12151.2 13525.9 10904.2 5885.8 1979.8 302.6

* Minimum value of the corresponding criterion.

Table 8: The WDisc, MDisc, CDisc, HDP and GWLP for the existing WDisc UDs on UD webpage. Source [76]

k	WDisc value	MDisc value	CDisc value	HDP	GWLP
3	0.100143524869*	0.108662634793*	0.032322467358*	0 81 162 108*	0 0 0*
4	0.179335170790*	0.234003148857*	0.046547389844*	0 0 162 108 81*	0 0 0 2 *
5	0.302610594748*	0.474581050924	0.063878130134	0 0 27 189 81 54*	0 0 2 6 0*
6	0.490921309445	0.925054072275	0.084316187127	0 0 0 72 189 54 36	0 0 5.3 14 4 2.7
7	0.777541070326	1.758082472968	0.109301381670	0 0 0 12 113 156 57 13	0 0 11.93 24.97 21.6 16.3 5.3
8	1.210094907473	3.287758405543	0.140337480437	0 0 0 0 32 148 124 44 3	0 0.1 19.8 46.4 62.1 63.7 41.2 8.8
9	1.848207900743	6.018653113763	0.174882736358	0 0 0 0 0 60 171 99 21 0	0 0 0.3 0.8 1.4 1.96 1.79 0.8 0.2
10	2.830672341649	11.041302123654	0.223909257557	0 0 0 0 0 8 95 174 65 8 1	0 0.01 0.43 1.36 2.77 4.98 5.87 4.16 1.91 0.38
11	4.277463587626	20.078640307248	0.285717465072	0 0 0 0 0 0 21 131 164 32 3 0	0 0.001 0.06 0.21 0.51 1.1 1.6 1.53 1.05 0.42 0.07
12	6.452832012611	36.226323326742	0.350849228465	0 0 0 0 0 0 1 39 183 107 21 0 0	0 0.002 0.08 0.31 0.89 2.18 3.82 4.65 4.14 2.54 0.89 0.15
13	9.679638830388	65.293241156762	0.454894947090	0 0 0 0 0 0 0 2 81 188 75 5 0 0	0 0.0002 0.01 0.045 0.15 0.41 0.82 1.21 1.35 1.09 0.59 0.2 0.03

* Minimum value of the corresponding criterion.

Table 9: The CDisc, MDisc, WDisc, HDP and GWLP for the existing recommended designs. Source [76]

k	CDisc value	MDisc value	WDisc value	HDP	GWLP
3	0.032322467358*	0.108662634793*	0.100143524869*	0 81 162 108*	0 0 0*
4	0.046547389844*	0.234003148857*	0.179335170790*	0 0 162 108 81*	0 0 0 2*
5	0.063688708041	0.474539911712	0.302610594748*	0 0 27 189 81 54*	0 0 2 6 0*
6	0.083475495916*	0.922923159349*	0.489790723912*	0 0 0 54 243 0 54*	0 0 4 18 0 4*
7	0.108060520606	1.754300617510	0.775286878183*	0 0 0 0 135 162 27 27*	0 0 10 30 18 16 6 *
8	0.136644427607*	3.262418935064	1.201012420711*	0 0 0 0 0 216 108 0 27*	0 0 16 60 48 64 48 6*
9	0.170995503086*	5.981605078476	1.835440251825*	0 0 0 0 0 0 324 0 0 27*	0 0 24 108 108 192 216 54 26*
10	0.213993691155*	10.921071686276	2.796660252959*	0 0 0 0 0 0 81 243 0 27 0*	0 0 42 144 270 480 630 378 206 36 *
11	0.264549304132*	19.710159154846	4.212689917623*	0 0 0 0 0 0 0 162 162 27 0 0 *	0 0 60 216 504 1092 1620 1530 1034 432 72*
12	0.325026925255*	35.294671843679	6.300095059932*	0 0 0 0 0 0 0 0 243 108 0 0 0*	0 0 80 324 864 2184 3888 4590 4136 2592 864 160*
13	0.397890251523*	62.882235480	9.381977149599*	0 0 0 0 0 0 0 0 0 351 0 0 0 0*	0 0 104 468 1404 4056 8424 11934 13442 11232 5616 2080 288*
13◇	0.426521643670	62.801081659727	9.381977149599*	0 0 0 0 0 0 0 0 0 351 0 0 0 0 *	0 0 104 468 1404 4056 8424 11934 13442 11232 5616 2080 288*

* Minimum value of the corresponding criterion.
◇ The recommended design by [59].

Table 10: The MDisc, CDisc, WDisc, HDP and GWLP for the new recommended designs via MDisc. Source [76]

k	Best MDisc value	The corresponding CDisc	The corresponding WDisc	The corresponding HDP	The corresponding GWLP
3	0.108662634793*	0.032322467358*	0.100143524869*	0 81 162 108*	0 0 0 0*
4	0.234003148857*	0.046547389844*	0.179335170790*	0 0 162 108 81*	0 0 0 2*
5	0.474281946451*	0.063335580099*	0.303078644335	0 3 24 180 96 48	0 0 2.7 5.3 0
6	0.922923159349*	0.083585957292	0.489790723912*	0 0 0 54 243 0 54*	0 0 4 18 0 4*
7	1.752757020660*	0.108184068537	0.776009196718	0 0 0 0 144 135 54 18	0 0 10.7 27.3 22 13.3 6.7
8	3.260443713968*	0.137380550265	1.201012420711*	0 0 0 0 0 216 108 0 27*	0 0 16 60 48 64 48 6*
9	5.978426686877*	0.172549264292	1.835440251825*	0 0 0 0 0 0 324 0 0 27*	0 0 24 108 108 192 216 54 26*
10	10.905672844220*	0.217586623274	2.796660252959*	0 0 0 0 0 0 81 243 0 27 0*	0 0 42 144 270 480 630 378 206 36 *
11	19.680022075626*	0.272522148327	4.212689917623*	0 0 0 0 0 0 0 162 162 27 0 0*	0 0 60 216 504 1092 1620 1530 1034 432 72*
12	35.242656253197*	0.344841878353	6.300095059932*	0 0 0 0 0 0 0 0 243 108 0 0 0*	0 0 80 324 864 2184 3888 4590 4136 2592 864 160*
13	62.798748642750*	0.426521643670	9.381977149599*	0 0 0 0 0 0 0 0 0 351 0 0 0 0*	0 0 104 468 1404 4056 8424 11934 13442 11232 5616 2080 288*

* Minimum value of the corresponding criterion.

Table 11: The CDisc, MDisc, WDisc, HDP and GWLP for the new recommended designs via CDisc. Source [76]

k	Best CDisc value	The corresponding MDisc	The corresponding WDisc	The corresponding HDP	The corresponding GWLP
3	0.032322467358*	0.108662634793*	0.100143524869*	0 81 162 108*	0 0 0*
4	0.046547389844*	0.234003148857*	0.179335170790*	0 0 162 108 81*	0 0 0 2*
5	0.063335580099*	0.474281946451*	0.303078644335	0 3 24 180 96 48	0 0 2.7 5.3 0
6	0.083475495916*	0.922923892677	0.489790723912*	0 0 0 54 243 0 54*	0 0 4 18 0 4*
7	0.108049122138*	1.755022026032	0.776009196718	0 0 0 0 144 135 54 18	0 0 10.7 27.3 22 13.3 6.7
8	0.136644427607*	3.262616194655	1.201012420711*	0 0 0 0 0 216 108 0 27*	0 0 16 60 48 64 48 6*
9	0.170995503086*	5.982975334405	1.835440251825*	0 0 0 0 0 0 324 0 0 27*	0 0 24 108 108 192 216 54 26*
10	0.213993691155*	10.922150287895	2.796660252959*	0 0 0 0 0 0 81 243 0 27 0*	0 0 42 144 270 480 630 378 206 36*
11	0.264549304132*	19.709161783471	4.212689917623*	0 0 0 0 0 0 0 162 162 27 0 0*	0 0 60 216 504 1092 1620 1530 1034 432 72*
12	0.325026925255*	35.306148647233	6.300095059932*	0 0 0 0 0 0 0 0 243 108 0 0 0*	0 0 80 324 864 2184 3888 4590 4136 2592 864 160*
13	0.397890251523*	62.946895856242	9.381977149599*	0 0 0 0 0 0 0 0 0 351 0 0 0 0*	0 0 104 468 1404 4056 8424 11934 13442 11232 5616 2080 288*

* Minimum value of the corresponding criterion.

Table 12: The CDisc, MDisc, WDisc, HDP and GWLP for the new recommended designs via WDisc. Source [76]

k	Best WDisc value	The corresponding MDisc	The corresponding CDisc	The corresponding HDP	The corresponding GWLP
5	0.302610594748*	0.474540524237	0.063714424286	0 0 27 189 81 54*	0 0 2 6 0*
7	0.775286878183*	1.754922399604	0.108466962713	0 0 0 0 135 162 27 27*	0 0 10 30 18 16 6*

* Minimum value of the corresponding criterion.

- A design is a minimum NBP design, a minimum χ^2P design, and a minimum DP design if and only if it sequentially minimizes any one of these criteria. A design is a minimum aberration design (MAD) if and only if it is a minimum OP design.
- Minimizing DDisc with parameters $a = \frac{3}{2}$ and $b = \frac{8q-1}{6q}$ is fully consistent with minimizing AWDisc. Thus, a design has minimum AWDisc iff it is uniform under DD with these parameters. Similarly, minimizing DDisc with $a = \frac{15q^2-3f(q)}{12q^2}$ and $b = \frac{13q^2-3f(q)-2q}{12q^2}$ is equivalent to minimizing ACDisc; and minimizing DDisc with $a = \frac{42q^2+3f(q)}{24q^2}$ and $b = \frac{38q^2+3f(q)-4q}{24q^2}$ is equivalent to minimizing AMDisc. Here, $f(q) = 1$ if q is even and $f(q) = 0$ if q is odd.
- The average discrepancies (AMDisc, ACDisc, AWDisc), GWLP, MAP, OP, NBP, DP, and χ^2P are approximately equivalent. Consequently, designs minimizing GWLP, MAP, OP, NBP, DP, or χ^2P tend to exhibit low average discrepancy.
- Generally speaking, all these criteria are closely related; some are equivalent in general or under certain conditions. An optimal design for screening quantitative factors may also be optimal for qualitative factors. For constructing a minimum aberration orthogonal nearly uniform array, the GWLP criterion is highly recommended. For constructing a minimum HDP nearly uniform minimum moment aberration design, the HDP criterion is strongly advised.

[84] extended the results of [83] from using only level permutation to employing the level permutation and factor projection (LPFP) algorithm. They examined the goodness of a full-dimensional design by evaluating all its possible projection sub-designs, providing an efficient benchmark for assessing optimality in lower dimensions. The relationships between full-dimensional designs and their projections are detailed in [84], with main findings as follows:

- A design is a minimum aberration full-dimensional design iff it is a minimum aberration projection design. Equivalently, a minimum aberration full-dimensional design is also a uniform projection design under any uniformity criterion.
- A full-dimensional orthogonal array is optimal iff it is an optimal projection orthogonal array under any orthogonality criterion. Moreover, an optimal full-dimensional orthogonal array under any orthogonality criterion is also a uniform projection design under any uniformity criterion.
- A full-dimensional uniform design is not guaranteed to be a uniform projection design. In such cases, uniform projection designs are highly recommended and generally superior.
- If a design is an optimal projection design under any one of the uniformity, aberration, or orthogonality criteria, it is also optimal under the others. Saturated orthogonal designs exhibit identical projection performance and cannot be distinguished via average projection.

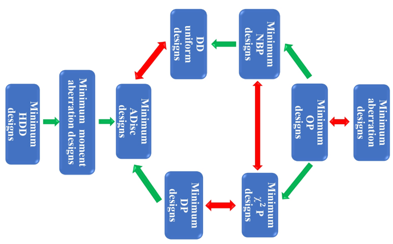


Figure 6: The bridges among designs, where ⇒ means "tends to" and ⇒ means "tends to (nearly)". Source [83]

3.6 Adjusted Criteria for Simplifying the Search Process

As discussed earlier, a variety of design criteria—including the GWLP, HDP, and many others—are characterized by multi-element patterns, each focusing on a different aspect of statistical inference for design comparison and selection. In practice, however, these patterns grow longer and more complex as designs incorporate ever-increasing numbers of factors and levels. This makes their representation and comparison inconvenient and computationally costly, creating a demand for simplifying such criteria.

[85] adopted the dictionary cross-entropy loss function (DCELF) from deep learning to convert any criterion pattern from a sequence to a scalar. The resulting scalar measure enables more straightforward and efficient design comparisons. The DCELF transforms a pattern X with m elements from a vector to a scalar as follows:

$$\mathrm{DCELF}(X) = \lambda + 1 - \frac{1-r}{r(1-r^{\lambda})} \sum_{i=1}^{\lambda} \frac{r^i}{\mathrm{CEL}(0, (x_{\mathrm{sub}})_i)},$$

where X_{sub} is the sub-pattern containing the first non-zero element and all subsequent elements in X, λ is the length of X_{sub}, $(x_{\mathrm{sub}})_i$ is the i^{th} element of X_{sub}, $r = 1 - \frac{\mathrm{CEL}(0,c)}{2\,\mathrm{CEL}(0,c)-1}$ is a ratio parameter, $\mathrm{CEL}(0,x) = -\frac{1}{2}\left[\log_2 f(x)(1-f(x))\right]$ is the cross-entropy loss function, $f(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function of element x, and c is a predefined threshold for X.

Table 13 (source [85]) presents a collection of GWLP values and their corresponding dictionary losses. In practice, the threshold for elements in GWLP can be set to $c = 0.0001$, and the first element of GWLP (which is always $A_0 = 1$) is omitted. The table illustrates that DCELF successfully reduces GWLP from a pattern to a scalar, while preserving consistency with dictionary ordering. For example, the sub-pattern $(X_1)_{\mathrm{sub}}$ derived from the first GWLP entry X_1 is $[7.5833, 19.4167, 17.3750, 11.4167, 3.9583]$. Here, $\lambda = 5$ and, with $c = 0.0001$, the ratio is $r = 1.8034 \times 10^{-9}$. Computing the binary cross-entropy loss for each element yields $\mathrm{DCELF}(X_1) = 5.8172$. Similarly, the sub-pattern for the last GWLP entry X_{12} is $(X_{12})_{\mathrm{sub}} = [6.0000, 40.0000, 60.0000, 66.0000, 58.0000, 12.0000]$, with length $\lambda = 6$ and the same ratio r. A similar calculation gives $\mathrm{DL}(X_{12}) = 6.7691$.

Table 13: The DCELF of various GWLP. Source [85]

No.	GWLP	DL(GWLP)
1	[1.0000 0.0000 0.0000 7.5833 19.4167 17.3750 11.4167 3.9583]	5.8172
2	[1.0000 0.0000 0.0000 11.9259 24.9630 21.5556 16.2963 5.2593]	5.8838
3	[1.0000 0.0000 0.0000 22.0000 34.5000 27.0000 31.0000 6.0000]	5.9370
4	[1.0000 0.0000 0.2143 4.2143 19.1020 15.7653 7.7347 4.0408]	6.0082
5	[1.0000 0.0000 0.2485 4.5325 21.7633 14.8757 9.4320 4.2249]	6.0110
6	[1.0000 0.0000 0.3471 6.3802 25.0413 16.7603 11.7355 5.0083]	6.0212
7	[1.0000 0.0000 0.4200 7.8400 25.9200 20.1000 11.6800 5.9400]	6.0306
8	[1.0000 0.0000 0.6562 11.8750 29.6250 24.9375 16.0938 6.9375]	6.0709
9	[1.0000 0.0000 0.8571 14.7755 32.6939 26.6939 21.1429 6.9796]	6.1140
10	[1.0000 0.0000 1.6800 23.9200 41.5200 34.3200 35.9200 7.4400]	6.3143
11	[1.0000 0.0000 2.6250 31.7500 48.0000 43.5000 47.1250 8.2500]	6.4986
12	[1.0000 0.0000 6.0000 40.0000 60.0000 66.0000 58.0000 12.0000]	6.7691

4 On the Construction of Uniform Fold-Over Experimental Designs

FrFDs are widely used for screening experiments. However, the cost we pay of using FrFDs is the alias structure of main effects or interactions. A common method to deal is to add more runs. A standard follow-up strategy discussed in many textbooks involves adding a second fraction, called a fold-over design. fold-over of a FrFD is a quick technique to create a design with twice as many runs, which typically releases aliased factors or interactions. A classical technique to fold-over two-level $(-,+)$ FrFDs is to reverse the plus and minus signs of one or more columns of the original design. The new fraction is called a fold-over design. A fold-over plan refers to the collection of columns (factors) whose signs are reversed in the fold-over design. A combined design is obtained by adding the fold-over design to the corresponding initial design. An optimal fold-over plan refers to a that can be used to construct an optimal combined design in view of a given criterion. This methodology successfully applied across diverse fields including agriculture, environmental science, computer experiments, and microarray analysis [86, 87, 88, 89, 90]. A notable application was demonstrated by [91], who used a sequential design to optimize supercritical CO_2 extraction parameters, successfully isolating the compounds emodin and physcion with an ethanol modifier.

Example 4.1 *Table 14 gives the regular FrFD in Table 2 with its fold-over design. From Table 14, we can get that: (i) There is a link between the main effect of F_3 and the interaction effect F_1F_2 in the original design (first half). (ii) There is a link between the main effect of F_3 and the interaction effect F_1F_2 in the fold-over design (second half). (iii) By combing the two half together in one design with 8 runs, there is no links between the effects. Moreover, we can see that the resulting combined design is the FuFD in Table 1.*

Several popular fold-over techniques, including the adjusted switching levels fold-over technique, rotating levels fold-over technique and permuting levels fold-over technique are reviewed in this section, where the general algorithm of these methods is also given.

- *Rotating levels fold-over technique* (T_R) [92]: For any initial design $\mathbf{d} \in \mathbb{U}_n(q^s)$, the basic idea is to add a number chosen from the set of levels, i.e. $\{0, 1, \cdots, q-1\}$, and then take the remainder to q for level of each factor. Let Γ be the

Table 14: Folding over a fractional factorial two-level design

The original fractional factorial design			
Runs No	F_1	F_2	$F_3 = F_1F_2$
r_1	-1	-1	+1
r_2	-1	+1	-1
r_3	+1	-1	-1
r_4	+1	+1	+1
The optimal fold-over plan			
	-1	-1	-1
The corresponding fold over design			
Runs No	$F_1^*(=-F_1)$	$F_2^*(=-F_2)$	$F_3^*(=-F_3) = -F_1^*F_2^*$ i.e. $F_3 = -F_1F_2$
r_5	+1	+1	-1
r_6	+1	-1	+1
r_7	-1	+1	+1
r_8	-1	-1	-1

collection of fold-over plans of $\mathbf{d}$, where $\Gamma = \{\Gamma_i = [\gamma_1, \gamma_2, \cdots, \gamma_s] | i = 1, 2, \cdots, N; \gamma_j = 0, 1, \cdots, q-1; 1 \le j \le s\}$. Each fold-over plan γ performs the operation on the original design $\mathbf{d} = (d_{ij})_{i=1,j=1}^{n,s}$ to obtain fold-over design $\mathbf{F} = (F_{ij})_{i=1,j=1}^{n,s}$, i.e.

$$F_{ij} = (d_{ij} + \gamma_j) \ mod \ q, \ 1 \le i \le n, \ 1 \le j \le s, \ 0 \le d_{ij} \le q-1$$

As the original design $\mathbf{d}$ has q levels and s factors, a larger search domain of optimal fold-over plan will be achieved where there are $N = q^s$ possible combinations of fold-over plans.

- *Permuting levels fold-over technique* (T_P) [93]: Consider a design $\mathbf{d} \in \mathbb{U}_n(q^s)$, the set of fold-over plan Γ can be defined in a similar manner, i.e., $\Gamma = \{\Gamma_i = [\gamma_1, \gamma_2, \cdots, \gamma_s] | i = 1, 2, \cdots, N\}$, where γ is a $q \times s$ matrix, $\gamma_j = [\gamma_{1j}, \gamma_{2j}, \cdots, \gamma_{qj}]'$ is the j-th column of γ which indicates a specific plan for j-th factor in $\mathbf{d}$ and takes value from the set of levels, i.e. $\{\gamma_{1j}, \gamma_{2j}, \cdots, \gamma_{qj}\} = \{0, 1, \cdots, q-1\}$, $\gamma_{ij} \in \{0, 1, \cdots, q-1\}$ specifies the mapping for the i-th level of j-th factor, $j = 1, 2, \cdots, s$. In essence, it transforms the original design $\mathbf{d} = (d_{ij})_{i=1,j=1}^{n,s}$ into fold-over design $\mathbf{F} = (F_{ij})_{i=1,j=1}^{n,s}$ based on the level permutations, i.e.,

$$F_{ij} = \gamma_{(d_{ij})j}, \ 1 \le i \le n, \ 1 \le j \le s, \ 0 \le d_{ij} \le q-1.$$

  Obviously, there are $q!$ choices for each column of γ, along with s factors it provides $N = (q!)^s$ fold-over plans which are in a much larger space than that of the other two methods for searching the optimal combined design.

- *Adjusted switching levels fold-over technique* (T_A) [85]: Similarly to the classical way that two levels are represented by inverse signs, levels of factors in multilevel U-type designs in $\mathbb{U}_n(q^s)$ are pairwise complementary, provided that the sum of levels are equal in pairs, e.g., $\{j, q-j-1\}, \ 0 \le j \le q-1$. Therefore, a sign-switched level will correspond to its complementary level. Given an initial design $\mathbf{d} \in \mathbb{U}_n(q^s)$, the fold-over design $\mathbf{F} = (F_{ij})_{i=1,j=1}^{n,s}$ of the initial design $\mathbf{d} = (d_{ij})_{i=1,j=1}^{n,s}$ is given by

$$F_{ij} = (1-\gamma_j)d_{ij} + \gamma_j(q-1-d_{ij}), \ 1 \le i \le n, \ 1 \le j \le s, \ 0 \le d_{ij} \le q-1$$

  where $\gamma_j = 0$ when the levels of the j-th factor remain the same and $\gamma_j = 1$, otherwise. Let Γ be the set of fold-over plans in $\mathbf{d}$, $\Gamma = \{\Gamma_i = [\gamma_1, \gamma_2, \cdots, \gamma_s] | i = 1, 2, \cdots, N; \gamma_j = 0, 1; 1 \le j \le s\}$, where N is the number of possible fold-over plans. Since each factor keeps or switches sign independently, it implies that the searching process will be $N = 2^s$ fold-over plans and corresponding combined designs. It is worth-mention that for two-level design this technique is the same as the classical technique.

Essentially, the above three techniques share the same structure and hence they can be summarized by the following algorithm (cf. Algorithm 1) for selecting the optimal fold-over plan (combined design) from the set of all the possible fold-over plans (combined designs) based on a given criterion.

Example 4.2 *[85] Let the initial design* $\mathbf{d} \in \mathbb{U}_8(4^6)$*, given in Table 15. The combined design is defined by* $\mathbf{C} = \begin{pmatrix} \mathbf{d} \\ \mathbf{F} \end{pmatrix}$. *There are* $2^6 = 128$, $4^6 = 4096$ *and* $(4!)^6 = 191102976$ *combined designs corresponding to the fold-over plans and fold-over designs using the above-mentioned three methods* T_A, T_R *and* T_P*, respectively. For the given fold-over plans*

Algorithm 1 The general fold-over algorithm. Source [85]

Input the original design $\mathbf{d} \in \mathbb{U}_n(q^s)$
Select an optimization criterion based on any optimization perspective as mentioned above
Generate the fold-over plan space $\mathit{\Gamma} = \bigcup_{i=0}^{N} \mathit{\Gamma}_i$ using any of the above-mentioned three techniques
for $i = 1$ to N **do**
    Generate the fold-over design $\mathbf{F}_i$ using any of the above-mentioned three techniques
    Generate the corresponding combined design $\mathbf{C}_i = \begin{pmatrix} \mathbf{d} \\ \mathbf{F}_i \end{pmatrix}$ for the current fold-over plan $\mathit{\Gamma}_i \in \mathit{\Gamma}$
    Calculate the value of the used optimization measure for the generated combined design $\mathbf{C}_i$
end for
Output the optimal combined design $\mathbf{C}^*$ by optimizing (minimizing or maximizing) the given criterion

$\Gamma_A = [0\ 1\ 0\ 1\ 0\ 1]$, $\Gamma_R = [0\ 2\ 1\ 2\ 3\ 0]$ and $\Gamma_P = \begin{bmatrix} 0 & 3 & 2 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 & 3 & 2 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 & 1 & 3 \end{bmatrix}$ *the fold-over design* $\mathbf{F}$ *via T_A, T_R and T_P are shown in Table 15.*

Table 15: The initial and fold-over designs for the Example 3.2. Source [85]

d						**F** via T_A						**F** via T_R						**F** via T_P					
2	3	1	3	3	1	2	0	1	0	3	2	2	1	2	1	2	1	2	2	3	0	1	2
3	0	2	2	2	2	3	3	2	1	2	1	3	2	3	0	1	2	3	3	0	1	0	0
0	2	0	2	1	1	0	1	0	1	1	2	0	0	1	0	0	1	0	1	2	1	3	2
0	3	3	1	3	3	0	0	3	2	3	0	0	1	0	3	2	3	0	2	1	2	1	3
3	1	3	3	0	0	3	2	3	0	0	3	3	3	0	1	3	0	3	0	1	0	2	1
1	1	1	1	2	0	1	2	1	2	2	3	1	3	2	3	1	0	1	0	3	2	0	1
1	2	2	0	0	2	1	1	2	3	0	1	1	0	3	2	3	2	1	1	0	3	2	0
2	0	0	0	1	3	2	3	0	3	1	0	2	2	1	2	0	3	2	3	2	3	3	3

The classical fold-over of two-level fractional factorial designs (FrFDs) has been extensively discussed in the literature, with the aberration criterion often serving as the optimality criterion [94, 95, 96, 97]. [98] examined optimal fold-over plans for two-level resolution IV designs by reversing the signs of one or more factors. [99] studied optimal fold-over plans for Plackett-Burman designs, the most widely used non-regular designs. [100] focused on optimal fold-over plans for two-level resolution III designs. [101] provided a complete collection of optimal fold-over plans for regular two-level designs with 16 or 32 runs. Subsequently, [102] further investigated optimal fold-over plans for non-regular orthogonal designs. [103] analyzed the theoretical properties of regular combined designs with two-level factors. More recently, [99] explored the use of non-orthogonal fold-over designs. [104] suggested that (1) uniformity criteria—which apply to both regular and non-regular designs—should be used as optimality criteria for constructing combined designs, and (2) the initial design itself should be a uniform design, offering a good representation of the experimental domain with fewer runs. A fold-over plan that yields a combined design with the smallest uniformity criterion value among all possible plans is termed an optimal fold-over plan. The fold-over of uniform designs has been widely examined; for example, [104] first employed the CDisc as a uniformity criterion to search for optimal fold-over plans. [105] derived lower bounds for the CDisc of combined designs when all Hamming distances between any pair of distinct runs in the two-level initial design are equal. [106] obtained lower bounds for the CDisc, symmetric L_2-discrepancy (SDisc), and WDisc for two-level combined designs in the general case.

4.1 Key Findings for Folding Over Two-Level Designs

[92] introduced the aforementioned rotating levels fold-over technique, which can be applied to designs with any number of levels. Using this technique, they investigated the theoretical properties of combined designs involving two-level factors, i.e., designs $\mathbf{d} \in \mathbb{U}_n(2^s)$, with respect to various uniformity criteria: LDisc, SDisc, CDisc, WDisc, and MDisc. An algorithm for searching for optimal fold-over plans was also developed (see Algorithm 2). As benchmarks for the new algorithm, they obtained new analytical expressions and lower bounds for these uniformity criteria for two-level combined designs. All of these discrepancies can be expressed in a single unified equation. A key finding is that a design $\mathbf{d} \in \mathbb{U}_n(2^s)$ admits the same optimal fold-over plan across all the discrepancies (LDisc, SDisc, CDisc, WDisc, and MDisc). Moreover, the results show that the new lower bounds derived with the new fold-over technique in [92] are more useful and tighter

than those obtained with the classical fold-over technique in [106] and the lower bound in [107] (see Figures 7, 8, and 9, source [92]).

Algorithm 2. Searching the optimal fold-over plans for a given design $\mathbf{d} \in \mathbb{U}_n(2^s)$. Source [92]

1: **Input** the original design $\mathbf{d} \in \mathbb{U}_n(2^s)$;

2: Generate the fold-over plan space $\Gamma = \bigcup_{t=0}^{s} \Gamma^{(t)}$; where $\Gamma^{(t)}$ is the set of fold-over plans $\Gamma_k^{(t)}$ with t non-zero elements

3: Set the current optimal fold-over plan $\Gamma^* = (0, \ldots, 0)$

4: Generate the fold-over design $\mathbf{F}^*$ and combined design $\mathbf{C}^* = \begin{pmatrix} \mathbf{d} \\ \mathbf{F}^* \end{pmatrix}$ for the current fold-over plan Γ^*

5: Set $Disc_0 = [Disc(\mathbf{C}^*]^2$;

6: **for** $t = 1$ to s **do**

7: **for** $k = 1$ to $|\Gamma_t|$ **do**

8: Compute $[Disc(\mathbf{C}_k^{(t)})]^2$, where $\mathbf{C}_k^{(t)}$ is the combined design via the k-th fold-over plan $\Gamma_k^{(t)}$;

9: **if** $[Disc(\mathbf{C}_k^{(t)})]^2 = LBDisc(t)$

10: **Renew** $\Gamma^* = \Gamma_k^{(t)}$;

11: **Renew** $Disc_0 = [Disc(\mathbf{C}_k^{(t)})]^2$;

12: **break** the current **for** loop

13: **elseif** $[Disc(\mathbf{C}_k^{(t)})]^2 < Disc_0$

14: **Renew** $\Gamma^* = \Gamma_k^{(t)}$;

15: **Renew** $Disc_0 = [Disc(\mathbf{C}_k^{(t)})]^2$;

16: **end if**

17: **end for**

18: **end for**

19: **Output** γ^* and $\mathbf{C}^*$.

4.2 Key Findings for Folding Over Three-Level Designs

[108] extended the work of [92] from two-level designs $\mathbf{d} \in \mathbb{U}_n(2^s)$ to three-level designs $\mathbf{d} \in \mathbb{U}_n(3^s)$. For three-level FrFDs as the original designs, fold-over plans and combined designs are analyzed in terms of uniformity criteria. Their results provide a theoretical justification for selecting optimal fold-over plans for three-level designs based on these criteria. The authors also defined the concept of the complementary fold-over plan as follows: for any fold-over plan $\Gamma = (\gamma_1, \ldots, \gamma_s) \in \Gamma^{(t)}$, the complementary fold-over plan is defined as

$$\Gamma^c = (3 - \gamma_1, \ldots, 3 - \gamma_s) \pmod 3 \in \Gamma^{(t)}$$

for any design $\mathbf{d} \in \mathbb{U}_n(3^s)$. The fold-over design and the combined design generated using Γ^c are called the complementary fold-over design $\mathbf{F}^c$ and the complementary combined design $\mathbf{C}^c$, respectively.

The main results show that the combined design $\mathbf{C}$ and its complementary combined design $\mathbf{C}^c$ yield the same LDisc and WDisc values, i.e.,

$$[\mathrm{LDisc}(\mathbf{C})]^2 = [\mathrm{LDisc}(\mathbf{C}^c)]^2, \qquad [\mathrm{WDisc}(\mathbf{C})]^2 = [\mathrm{WDisc}(\mathbf{C}^c)]^2.$$

Consequently, if $\Gamma \in \Gamma^{(t)}$ is an optimal fold-over plan for a three-level design $\mathbf{d} \in \mathbb{U}_n(3^s)$ under LDisc or WDisc, then its complementary fold-over plan $\Gamma^c \in \Gamma^{(t)}$ is also optimal for $\mathbf{d}$. Hence, there exist multiple optimal fold-over plans in the same set $\Gamma^{(t)}$ (at least two: the optimal plan and its complement) that produce combined designs with the smallest discrepancy under LDisc or WDisc.

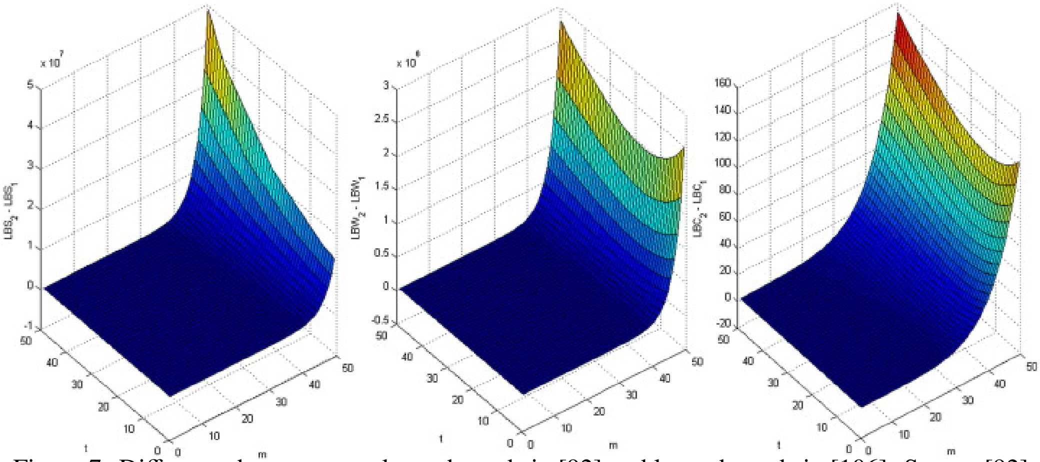


Figure 7: Difference between new lower bounds in [92] and lower bounds in [106]. Source [92]

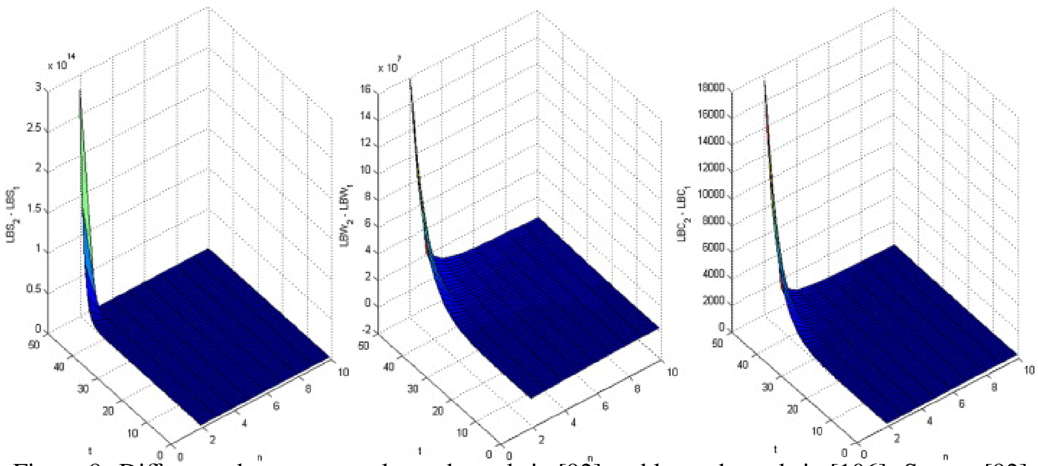


Figure 8: Difference between new lower bounds in [92] and lower bounds in [106]. Source [92]

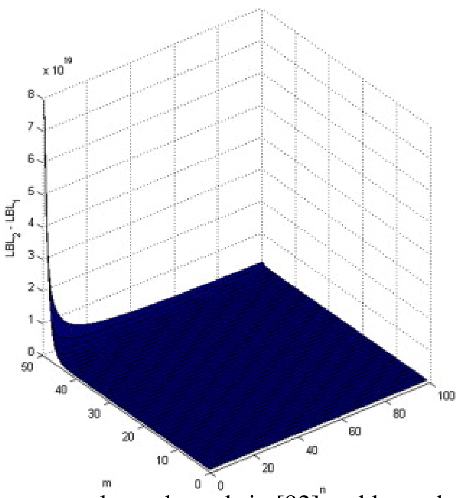


Figure 9: Difference between new lower bounds in [92] and lower bound in[107]. Source [92]

This finding significantly reduces computational complexity: searching for an optimal fold-over plan under LDisc or WDisc now requires only $(3^s+1)/2$ evaluations for a design $\mathbf{d} \in \mathbb{U}_n(3^s)$, compared to an exhaustive search of 3^s evaluations under CDisc and MDisc. Moreover, the results indicate that the optimal fold-over plan under LDisc is also optimal under WDisc, and vice versa, whereas CDisc exhibits different behavior (see Figures 10 and 11, source [108]). Lower bounds for the discrepancies of combined designs under general fold-over plans were also derived, providing benchmarks for searching optimal fold-over plans (combined designs) as outlined in Algorithm 2. Furthermore, the new lower bounds presented in [108] are more useful and sharper than those obtained using the classical fold-over technique in [109], as illustrated in Figure 12 (source [108]).

4.3 Key Findings for Folding Over Mixed Two- and Three-Level Designs

[110] extended the previous work in [92] and [108] to asymmetric designs with a mixture of two- and three-level factors, i.e., $\mathbf{d} \in \mathbb{U}_n(2^{s_1}3^{s_2})$. They examined the relationship between any initial design and its combined design, and provided a comparative study of combined designs under different discrepancy measures. The equivalence between a combined design and its complementary combined design was also investigated—a useful constraint that substantially reduces the search space. Using these results as benchmarks, one can implement a powerful algorithm for constructing optimal combined designs following Algorithm 2. This work not only encompasses but also improves upon results from approximately 20 recent articles as special cases, making it a valuable reference for constructing effective designs.

The main results demonstrate that for any mixed two- and three-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1}3^{s_2})$, the combined design $\mathbf{C}$ and its complementary combined design $\mathbf{C}^c$ share the same LDisc and WDisc values:

$$[\text{LDisc}(\mathbf{C})]^2 = [\text{LDisc}(\mathbf{C}^c)]^2, \qquad [\text{WDisc}(\mathbf{C})]^2 = [\text{WDisc}(\mathbf{C}^c)]^2.$$

Consequently, if $\Gamma \in \Gamma^{(t)}$ is an optimal fold-over plan for such a design under LDisc or WDisc, its complementary fold-over plan $\Gamma^c \in \Gamma^{(t)}$ is also optimal. This finding greatly reduces computational complexity: under LDisc or WDisc, the search now requires only $2^{s_1} \times \frac{3^{s_2}+1}{2}$ evaluations, compared to the exhaustive $2^{s_1} \times 3^{s_2}$ evaluations needed under CDisc or MDisc. Moreover, the results show that the optimal fold-over plan under LDisc is also optimal under WDisc, and vice versa.

4.4 Key Findings for Folding Over Four-Level Designs

[111] extended the earlier work of [92] and [108] to four-level designs $\mathbf{d} \in \mathbb{U}_n(4^s)$. [111] examined the optimality of fold-over plans for four-level designs using the most common criteria: the GWLP, LDisc, WDisc, CDisc, and MDisc. The authors proved that LDisc, WDisc, and GWLP are equivalent in three respects: between any initial design and its corresponding fold-over design; between any fold-over design and its complementary fold-over design; and between any combined design and its complementary combined design. That is,

$$[\text{LDisc}(\mathbf{d})]^2 = [\text{LDisc}(\mathbf{F})]^2, \quad [\text{WDisc}(\mathbf{d})]^2 = [\text{WDisc}(\mathbf{F})]^2, \quad \text{GWLP}(\mathbf{d}) = \text{GWLP}(\mathbf{F}),$$

$$[\text{LDisc}(\mathbf{F})]^2 = [\text{LDisc}(\mathbf{F}^c)]^2, \quad [\text{WDisc}(\mathbf{F})]^2 = [\text{WDisc}(\mathbf{F}^c)]^2, \quad \text{GWLP}(\mathbf{F}) = \text{GWLP}(\mathbf{F}^c),$$

$$[\text{LDisc}(\mathbf{C})]^2 = [\text{LDisc}(\mathbf{C}^c)]^2, \quad [\text{WDisc}(\mathbf{C})]^2 = [\text{WDisc}(\mathbf{C}^c)]^2, \quad \text{GWLP}(\mathbf{C}) = \text{GWLP}(\mathbf{C}^c).$$

These equivalences, however, do not necessarily hold for CDisc and MDisc. Consequently, if $\Gamma \in \Gamma^{(t)}$ is an optimal fold-over plan for a four-level design $\mathbf{d} \in \mathbb{U}_n(4^s)$ under LDisc or WDisc, its complementary fold-over plan $\Gamma^c \in \Gamma^{(t)}$ is also optimal. This result substantially reduces computational complexity: a search under LDisc or WDisc now requires only $(4^s+1)/2$ evaluations, compared to the exhaustive 4^s evaluations required under CDisc or MDisc. The authors further noted that if the initial design is uniform (or has minimum aberration), then all 4^s corresponding fold-over designs are also uniform (or have minimum aberration). Nevertheless, only a few of these—at least the optimal plan and its complement—can serve as optimal fold-over designs. Thus, combining any two uniform (or minimum aberration) designs does not guarantee a uniform (or minimum aberration) combined design.

In addition, new analytical expressions and lower bounds for these discrepancies were derived for both initial and combined designs, providing useful benchmarks for constructing uniform designs. To illustrate the practical application of these theoretical results, the authors compiled a catalog of optimal fold-over plans for constructing uniform minimum aberration symmetric four-level combined designs with $2 \leq s \leq 10$ factors and $8 \leq n \leq 52$ runs. These plans are suitable for experiments involving either qualitative or quantitative factors. Tables 16 and 17 (source [111]) list the optimal fold-over plans for uniform designs $\mathbf{d} \in \mathbb{U}_n(4^s)$ taken from the uniform-design website `https://www.math.hkbu.edu.hk/UniformDesign/` for $2 \leq s \leq 10$ factors and $8 \leq n \leq 52$ runs. The first two columns of each table give the number of factors s and runs n of the initial uniform design. The sixth and seventh columns present one optimal fold-over plan $\mathbf{f}^*$ (selected via WDisc and GWLP) and its complementary optimal fold-over plan $\mathbf{f}^{*c}$, respectively. The third column shows the squared WDisc values of the optimal combined design $\mathbf{C}^*$ and its complementary version $\mathbf{C}^{*c}$. The fourth and

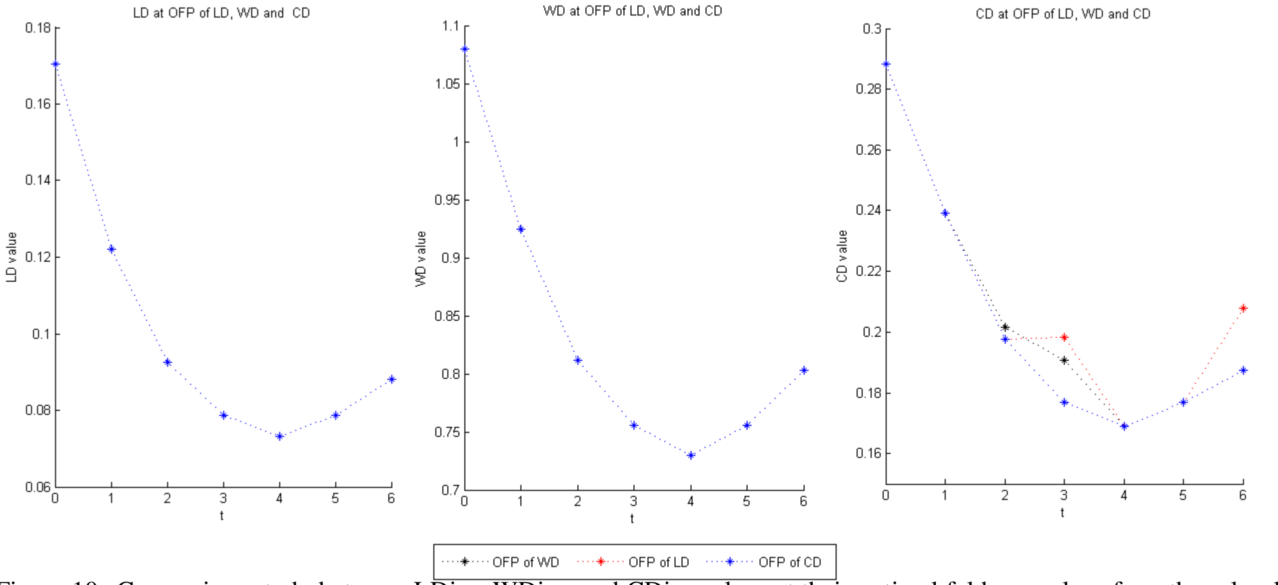


Figure 10: Comparison study between LDisc, WDisc, and CDisc values at their optimal fold-over plans for a three-level design with 6 factors. Source [108]

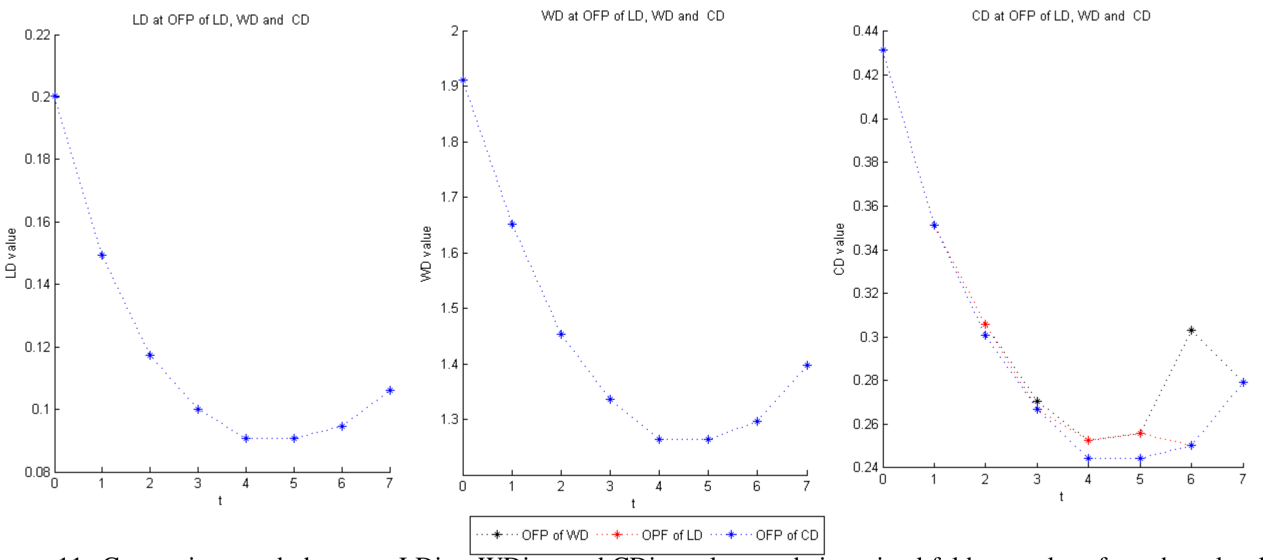


Figure 11: Comparison study between LDisc, WDisc, and CDisc values at their optimal fold-over plans for a three-level design with 7 factors. Source [108]

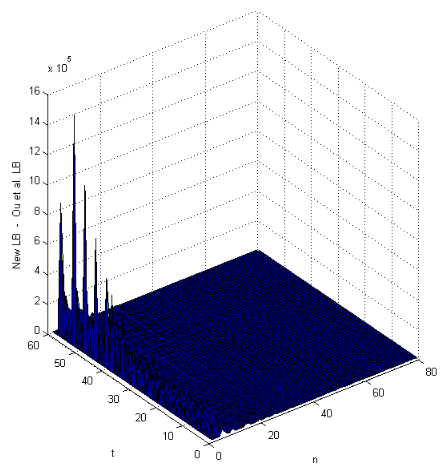


Figure 12: Difference between new lower bounds in [108] and lower bound in [109]. Source [108]

fifth columns display the two new lower bounds from [111] for the optimal combined design. The eighth column lists the GWLP of the initial design, any fold-over design, and the corresponding complementary fold-over design. Finally, the last column provides the GWLP of the optimal combined design and its complementary optimal combined design.

Table 16: Optimal foldover plans for the uniform designs $\mathbf{d} \in \mathbb{U}_n(4^s)$ for $2 \le s \le 5$. Source [111]

s	n	$[WDisc(\mathbf{C}^*)]^2 = [WDisc(\mathbf{C}^{*c})]^2$	LBW_1	LBW_2	$\mathbf{f}^*$	$\mathbf{f}^{*c}$	$GWLP(\mathbf{d}) = GWLP(\mathbf{F}^*) = GWLP(\mathbf{F}^{*c})$	$GWLP(\mathbf{C}^*) = GWLP(\mathbf{C}^{*c})$
2	8⋆	0.0279	0.0234	0.0291	0 1	0 3	(0, 1)	(0,0)
	12⋆	0.0281	0.0217	0.0279	0 2	0 2	(0, 0.3333)	(0,0.1111)
	16⋆	0.0279	0.0211	0.0276	1 1	3 3	(0,0)	(0,0)
	20⋆	0.0280	0.0208	0.0275	0 2	0 2	(0,0.12)	(0, 0.04)
	24⋆	0.0279	0.0207	0.0276	0 1	0 3	(0, 0.1111)	(0,0)
	28⋆	0.0279	0.0204	0.0273	0 2	0 2	(0, 0.0612)	(0, 0.0204)
	32⋆	0.0279	0.0203	0.0275	1 0	3 0	(0,0)	(0,0)
	36⋆	0.0280	0.0202	0.0272	0 2	0 2	(0,0.0370)	(0, 0.0123)
	40⋆	0.0279	0.0207	0.0278	0 1	0 3	(0, 0.04)	(0, 0)
	44⋆	0.0279	0.0200	0.0272	0 2	0 2	(0, 0.0248)	(0, 0.0083)
	48⋆	0.0279	0.0205	0.0276	0 0	0 0	(0, 0)	(0, 0)
	52⋆	0.0279	0.0204	0.0276	0 2	0 2	(0, 0.0178)	(0, 0.0059)
3	8⋆	0.0578	0.0508	0.0554	0 0 1	0 0 3	(0, 3, 4)	(0, 1, 2)
	12⋆	0.0573	0.0459	0.0520	0 2 2	0 2 2	(0, 1, 3.3333)	(0, 0.3333, 1.3333)
	16⋆	0.0561	0.0458	0.0518	0 0 1	0 0 3	(0, 0, 3)	(0, 0, 1)
	20⋄	0.0565	0.0430	0.0490	2 2 0	2 2 0	(0, 0.36, 1.84)	(0, 0.12, 0.8)
	20•	0.0566	-	-	2 2 2	2 2 2	-	(0, 0.12, 0.48)
	24⋆	0.0562	0.0435	0.0494	0 0 1	0 0 3	(0, 0.3333, 1.3333)	(0, 0.1111, 0.4444)
	28⋆	0.0562	0.0418	0.0477	2 2 0	2 2 0	(0, 0.1837, 1.1020)	(0, 0.0612, 1.0612)
	32⋆	0.0560	0.0423	0.0481	1 0 0	3 0 0	(0, 0, 1)	(0, 0, 0)
	36⋆	0.0561	0.0412	0.0469	2 2 0	2 2 0	(0, 0.1111, 0.6667)	(0, 0.0370, 0.6420)
	40⋆	0.0561	0.0416	0.0474	1 0 0	3 0 0	(0, 0.12, 0.48)	(0, 0.0400, 0.1200)
	44⋆	0.0561	0.0413	0.0469	2 2 2	2 2 2	(0, 0.0744, 0.3802)	(0, 0.0248, 0.0992)
	48⋆	0.0560	0.0422	0.0781	0 1 0	0 3 0	(0, 0, 0.5556)	(0, 0, 0.111)
	52⋆	0.0561	0.0408	0.0465	0 2 2	0 2 2	(0, 0.0533, 0.1775)	(0, 0.0178, 0.0710)
4	8⋆	0.1041	0.0972	0.1013	2 0 2 2	2 0 2 2	(0, 6, 16, 9)	(0, 1, 10, 4)
	12⋄	0.1048	0.0871	0.0916	0 2 0 2	0 2 0 2	(0, 2, 13.3333, 5)	(0, 0.8889, 5.3333, 3.4444)
	12•	0.1049	-	-	0 2 2 2	0 2 2 2	-	(0, 0.6667, 6.6667, 2.3333)
	16⋆	0.1006	0.0825	0.0861	2 2 0 0	2 2 0 0	(0, 0, 12, 3)	(0, 0, 4, 3)
	20⋆	0.1017	0.0798	0.0822	2 2 2 0	2 2 2 0	(0, 0.72, 7.36, 3.72)	(0, 0.24, 3.2, 1.96)
	24⋆	0.1006	0.0790	0.0819	1 1 0 0	3 3 0 0	(0, 0.6667, 5.3333, 3.6667)	(0, 0.2222, 1.7778, 2.3333)
	28⋄	0.1008	0.0767	0.0790	0 0 2 2	0 0 2 2	(0, 0.3673, 4.4082, 3.3673)	(0, 0.1663, 1.551, 2.1837)
	28•	0.1009	-	-	2 0 2 2	2 0 2 2	-	(0, 0.1224, 2.0408, 1.4082)
	32⋆	0.1000	0.0758	0.0778	2 2 0 0	2 2 0 0	(0, 0, 4, 3)	(0, 0, 1, 2)
	36⋆	0.1005	0.0751	0.0766	0 2 2 2	0 2 2 2	(0, 0.2222, 2.6667, 3.2222)	(0, 0.0741, 1.2840, 1.1975)
	40⋆	0.1002	0.0750	0.0769	1 0 1 0	3 0 3 0	(0, 0.24, 2.32, 2.84)	(0, 0.08, 0.8, 1.64)
5	8⋆	0.1830	0.1741	0.1816	0 2 0 2 2	0 2 0 2 2	(0, 10, 40, 45, 32)	(0, 3, 21, 24, 15)
	12⋄	0.1778	0.1533	0.1590	2 2 2 0 0	2 2 2 0 0	(0, 4.2222, 30.6667, 27.6667, 21.7778)	(0, 1.5556, 14.2222, 15.2222, 10.6667)
	12•	0.178	-	-	2 2 2 2 2	2 2 2 2 2	-	(0, 1.1111, 14.6667, 15.666,7 10.2222)
	16⋄	0.1700	0.1465	0.1499	2 2 0 0 0	2 2 0 0 0	(0, 0, 30, 15, 18)	(0, 0, 14, 7, 10)
	16•	0.1701	-	-	0 2 0 2 2	0 2 0 2 2	-	(0, 0, 10, 15, 6)
	20⋄	0.1721	0.1378	0.1434	2 2 0 0 2	2 2 0 0 2	(0, 1.2, 18.4, 18.6, 12)	(0, 0.48, 7.6, 10.36, 6.16)
	20•	0.1726	-	-	2 2 2 0 2	2 2 2 0 2	-	(0, 0.4, 8.96, 9.16, 6.08)
	24⋆	0.1696	0.1386	0.1449	1 1 0 0 0	3 3 0 0 0	(0 1.1111 13.3333 18.3333 8.8889)	(0, 0.4444, 5.3333, 9.6667, 4.8889)
	28⋄	0.1698	0.1314	0.1385	2 2 0 0 2	2 2 0 0 2	(0, 0.6531, 11.2245, 16.3061, 7.3878)	(0, 0.2449, 4.6939, 9.0408, 3.3061)
	28•	0.1701	-	-	2 2 0 2 2	2 2 0 2 2	-	(0, 0.2041, 5.551, 8.102, 3.4286)
	32⋆	0.1679	0.1296	0.1373	2 0 2 0 1	2 0 2 0 3	(0, 0, 10, 15, 6)	(0, 0, 3, 9, 3)
	36⋄	0.1689	0.1279	0.1358	0 0 2 2 2	0 0 2 2 2	(0, 0.3704, 7.8519, 13.7407, 5.4815)	(0, 0.1481, 3.2840, 7.1481, 2.6420)
	36•	0.1690	-	-	2 0 2 2 2	2 0 2 2 2	-	(0, 0.1235, 3.6543, 6.4815, 2.9630])

⋆ U-uniform minimum aberration combined design.

⋄ U-uniform nearly minimum aberration combined design.

• U-minimum aberration nearly uniform combined design.

Table 17: Optimal foldover plans for the uniform designs $\mathbf{d} \in \mathbb{U}_n(4^s)$ for $6 \le s \le 10$. Source [111]

s	n	$[WDisc(\mathbf{C}^*)]^2 = [WDisc(\mathbf{C}^{*c})]^2$	LBW_1	LBW_2	$\mathbf{f}^*$	$\mathbf{f}^{*c}$	$GWLP(\mathbf{d}) = GWLP(\mathbf{F}^*) = GWLP(\mathbf{F}^{*c})$	$GWLP(\mathbf{C}^*) = GWLP(\mathbf{C}^{*c})$
6	$8^{\diamond}$	0.3206	0.3057	0.3159	022200	022200	(0, 17, 72, 147, 184, 91)	(0, 9, 32, 75, 96, 43)
	$8^{\bullet}$	0.3237	-	-	100112	300332	-	(0, 5.75, 38, 76.5, 88, 46.75)
	$12^{\diamond}$	0.2965	0.2613	0.2736	200222	200222	(0, 6.7778, 59.5556, 85.6667, 128.8889, 59.4444)	(0, 2.5556, 30.2222, 44.3333, 60.4444, 32.1111)
	$12^{\bullet}$	0.3001	-	-	202212	202232	-	(0, 2.2222, 36, 43.2222, 60.8889, 27.3333)
	$16^{\star}$	0.2769	0.2445	0.2611	020012	020032	(0, 3, 48, 63, 96 , 45)	(0, 0, 24, 33, 48, 22)
	$20^{\diamond}$	0.2804	0.2301	0.2484	220220	220220	(0, 2.04, 36.16, 56.28, 72, 37.32)	(0, 0.76, 16.96, 28.76, 36.16, 18.76)
	$20^{\bullet}$	0.2813	-	-	222022	222022	-	(0, 0.68, 18.56, 26.36, 36.48, 19.32)
	$24^{\diamond}$	0.2758	0.2229	0.2435	112001	332003	(0, 1.6667, 28.6667, 49, 59.3333, 31)	(0, 0.5417, 13.5556, 24.8611, 29.4444, 15.9306)
	$24^{\bullet}$	0.2759	-	-	201011	3203033	-	(0, 0.50, 13.6111, 25.6111, 28.8333, 15.7778)
	$28^{\diamond}$	0.2756	0.2172	0.2375	022220	022220	(0, 1, 23.6735, 45.1224, 48.1633, 27.3265)	(0, 0.3878, 10.8571, 23.5714, 22.449, 14.8776)
	$28^{\bullet}$	0.2758	-	-	022222	022222	-	(0, 0.3061, 12.8980, 21.5306, 24.6531, 12.7551)
	$32^{\diamond}$	0.2719	0.2181	0.2388	002200	002200	(0, 0, 22.375, 37.875, 43.125, 23.6250)	(0, 0, 10.75, 18.75, 20.25, 13.25)
	$32^{\bullet}$	0.2722	-	-	011100	033300	-	(0, 0, 9, 19.75, 23.5, 10.75)
	$36^{\diamond}$	0.2730	0.2102	0.2315	022022	022022	(0, 0.6296, 16.6914, 38.2099, 35.6543, 21.5926)	(0, 0.2346, 7.4568, 19.8889, 17.037, 11.2716)
	$36^{\bullet}$	0.2732	-	-	022222	022222	-	(0, 0.1852, 8.642, 18.8025, 18.2222, 10.037)
7	$8^{\diamond}$	0.5249	0.5112	0.5249	2020220	2020220	(0, 21, 140, 315, 672, 623, 276)	(0, 7, 74, 159, 332, 313, 138)
	$8^{\bullet}$	0.5687	-	-	2222222	2222222	-	(0, 7, 70, 175, 308, 329, 134)
	$12^{\star}$	0.4716	0.4348	0.4534	2222200	2222200	(0, 11, 96.6667, 215, 436, 423.6667, 182)	(0, 3, 49.5556, 110.1111, 217.3333, 209.4444, 92.2222)
	$16^{\star}$	0.4413	0.4172	0.4335	2222221	2222223	(0, 6.5, 72.5, 170, 313, 326.5, 134.5)	(0, 0.75, 36.75, 90.5, 151.5, 163.75, 67.75)
	$20^{\diamond}$	0.4454	0.3945	0.4124	2222222	2222222	(0, 3.72, 59.68, 137.08, 247.68, 262.68, 107.36)	(0, 1.48, 28.8, 69.88, 121.92, 133.72, 52.8)
	$20^{\bullet}$	0.4473	-	-	1330033	3110011	-	(0, 1.32, 30.32, 72.44, 118.24, 133.24, 53.04)
	$24^{\diamond}$	0.4360	0.3735	0.3917	2222122	2222322	(0, 2.4444, 48.7778, 118.7778, 201.5556, 220.8889, 89.2222)	(0, 0.7778, 22.8889, 61.6667, 99.5556, 110.7778, 44.6667)
	$24^{\bullet}$	0.4379	-	-	22201322	2220312	-	(0, 0.6667, 25, 58.7778, 101.1111, 112, 42.7778)
	$28^{\diamond}$	0.4352	0.3501	0.3704	2022220	2022220	(0, 1.5714, 40.5714, 107, 166.8571, 192.1429, 76)	(0, 0.5918, 19.3469, 53.1224, 84.2449, 96.7959, 37.4694)
	$28^{\bullet}$	0.4356	-	-	2022222	2022222	-	(0, 0.5510, 19.8776, 54.6735, 84.6531, 94.6327, 37.1837)
	$32^{\diamond}$	0.4295	0.3455	0.3664	2020002	2020002	(0, 0.4688, 37.9063, 88.6875, 152.8125, 163.3438, 67.7813)	(0, 0.1875, 17.9375, 44.3750, 76.3750, 82.4375, 33.6875)
	$32^{\bullet}$	0.4297	-	-	0103133	0301311	-	(0, 0.1328, 17.4922, 45.5781, 75.4844, 82.6641, 33.6484)
	$36^{\diamond}$	0.4300	0.3364	0.3565	2020012	2020032	(0, 0.8765, 30.0494, 85.7160, 130.1235, 147.4691, 59.8765)	(0, 0.3827, 13.5802, 44.0370, 65.3827, 72.1111, 31.0617)
	$36^{\bullet}$	0.4304	-	-	2222220	2222220	-	(0, 0.2840, 14.9383, 42.0617, 66.9136, 73.1975, 29.1605)
8	$8^{\diamond}$	0.8844	0.8466	0.8592	22321000	22123000	(0, 30, 212, 660, 1752, 2522, 2196, 819)	(0, 12, 106, 340, 868, 1256, 1106, 407)
	$8^{\bullet}$	0.8879	-	-	10220122	30220322	-	(0, 10.5, 111, 337.5, 858, 1273.5, 409.5)
	$12^{\diamond}$	0.7779	0.7026	0.7160	32111001	12333003	(0, 0.0164, 0.1449, 0.4522, 1.1360, 1.7124, 1.4498, 0.5486)	(0, 6.2222, 73.3333, 226.4444, 569.7778, 856, 722.2222, 275.6667)
	$12^{\bullet}$	0.7836	-	-	00222121	002223203	-	(0, 6.0556, 73.2222, 231.2778, 558.4444, 864.6111, 721.2222, 274.8333)
	$16^{\diamond}$	0.7147	0.6515	0.6700	22210222	22230222	(0, 9.2, 113.5, 343.8, 833, 1304.8, 1077.5, 413.3)	(0, 2.5, 56.5, 176, 417, 646.5, 542.5, 206)
	$16^{\bullet}$	0.7150	-	-	22220122	22220322	-	(0, 2.375, 58.75, 170.625, 418.5, 653.625, 534.75, 208.375)
	$20^{\diamond}$	0.7003	0.5960	0.6101	22003012	22001032	(0, 5.7, 92.6, 277.6, 660.8, 1046.8, 861.9, 330.4)	(0, 2.2, 45.28, 140.36, 330.88, 518.6, 437.92, 162.16)
	$20^{\bullet}$	0.7006	-	-	10001331	30003113	-	(0, 2.06, 44.36, 142.58, 328.88, 522.58, 432.52, 164.42)
	$24^{\diamond}$	0.6814	0.5699	0.5824	21022010	23022030	(0, 3.5556, 77.7778, 235.3333, 543.1111, 878.2222, 716, 275.6667)	(0, 1.1806, 38.75, 117.2083, 272.0556, 436.7083, 363.4167, 135.0139)
	$24^{\bullet}$	0.6825	-	-	21122010	23322030	-	(0, 1.0694, 38.75, 117.7639, 271.1667, 441.2639, 355.4167, 138.9028)
	$28^{\diamond}$	0.6732	0.5762	0.5921	22222222	22222222	(0, 2.3265, 65.5510, 207.3469, 460.5714, 751.7959, 616.7347, 235.2449)	(0, 0.8163, 30.6122, 107.2245, 229.7143, 374.3673, 307.8367, 118.7143)
	$28^{\bullet}$	0.6754	-	-	22102223	22302221	-	(0, 0.8061, 32.0612, 104.3776, 231.3878, 372.0918, 311.5714, 116.9898)
	$32^{\diamond}$	0.6663	0.5414	0.5528	10200202	30200202	(0, 1.2188, 58.6875, 180.2813, 403.6250, 658.2813, 538.6875, 206.2188)	(0, 0.5469, 26.6563, 92.3906, 202.4375, 329.5781, 267.4063, 103.9844)
	$32^{\bullet}$	0.6674	-	-	01103310	03301130	-	(0, 0.4297, 27.4844, 91.7578, 203.0313, 327.0703, 269.7344, 103.4922)
	$36^{\diamond}$	0.6645	0.5266	0.5360	20002122	20002322	(0, 1.1358, 50.1728, 162.6420, 361.8765, 577.5309, 483.9506, 182.1358)	(0, 0.4691, 22.3210, 83.5556, 183.1111, 285.5556, 243.3580, 90.8519)
	$36^{\bullet}$	0.6654	-	-	20022122	20022322	-	(0, 0.4198, 24, 81.4321, 182.9136, 286, 243.4568, 91)
9	$8^{\diamond}$	1.4140	1.3736	1.3977	100212220	300232220	(0 42 294 1260 3822 7686 9810 7395 2458)	(0, 16.5, 151.5, 638.5, 1895.5, 3845.5, 4914.5, 3690.5, 1230.5)
	$8^{\bullet}$	1.4249	-	-	103211120	301233320	-	(0, 16, 153, 640, 1883, 3868, 4895, 3699, 1229)
	$12^{\diamond}$	1.2301	1.1244	1.1542	012022220	032022220	(0, 22.7, 210.7, 822, 2562.7, 5110.7, 6552, 4923.7, 1640)	(0, 9.3, 104, 414.9, 1288.9, 2540.9, 3279.1, 2466.8, 817.8)
	$12^{\bullet}$	1.2356	-	-	112231001	332213003	-	(0, 8.4, 108.2, 413.6, 1274, 2560.4, 822)
	$16^{\diamond}$	1.1243	1.0058	1.0332	132020102	312020302	(0, 012.6, 166.1, 628.1, 1863.6, 3919.9, 4848.4, 3718.4, 1225.9)	(0, 4.3, 82.5, 316.9, 937.7, 1947.8, 2429.6, 186, 612.2)
	$16^{\bullet}$	1.1249	-	-	112010102	332030302	-	(0, 4.2, 82.6, 318.3, 933.7, 1950.2, 2432.3, 1856.3, 613.4)
	$20^{\star}$	1.0742	0.9609	0.9837	221022222	223022222	(0, 8.6, 132.6, 509.5, 1485.6, 3126.4, 3896.2, 2964.4, 982.9)	(0, 2.5, 67.9, 252.5, 746, 1562.8, 1949.2, 1478.5, 493.2)
	$24^{\diamond}$	1.0517	0.8892	0.9076	001212022	003232022	(0, 5.3 113.3 426, 1233.3, 2605.3, 3252, 2466.3, 820)	(0, 2.1, 55.7, 214.7, 616.5, 1299.3, 1629.5, 1234, 408.5)
	$24^{\bullet}$	1.0560	-	-	122032120	322012320	-	(0, 1.9, 59, 214.1, 613, 1305.8, 1619.5, 1232.8, 414.2)
	$28^{\diamond}$	1.0344	0.8573	0.8732	301212200	103232200	(0, 3.2, 97.5, 373.2, 1040.4, 2247.2, 2782.7, 2113.9, 703.1)	(0, 1.2, 47.5, 188.7, 520.3, 1116.6, 1399.4, 1056.6, 349.8)
	$28^{\bullet}$	1.0365	-	-	221202220	223202220	-	(0, 1.2, 47.4, 184.6, 525.1, 1124.7, 1390.5, 1052.6, 354.1)
	$32^{\star}$	1.0176	0.8357	0.8552	301010321	103030123	(0, 2.1, 85.6, 326.6, 914.1, 1958.4, 2440.9, 1847.9, 615.4)	(0, 0.6, 40.5, 167.4, 455.9, 974.8, 1225.8, 923.2, 306.8)
	$36^{\diamond}$	1.0100	0.8467	0.8672	222222222	222222222	(0, 2, 73.2, 292.7, 825.8, 1708.5, 2200.4, 1628.8, 549.4)	(0, 0.8, 33.1, 150.9, 411.1, 854.8, 1102.9, 809.7, 276.5)
	$36^{\bullet}$	1.0125	-	-	222221202	222223202	-	(0, 0.7, 35.8, 145.2, 415, 853, 1103.2, 811.9, 275.1)
10	$8^{\star}$	2.2255	2.2230	2.2255	2200202222	2200202222	(0, 51, 432, 2058, 7728, 19110, 32784, 36933, 24592, 7383)	(0, 19, 224, 1050, 3808, 9590, 16416, 18421, 12320, 3687)
	$12^{\diamond}$	1.9188	1.7865	1.8282	2200230102	2200210302	(0, 28, 300, 1386, 5071, 12872, 21741, 24679, 16380, 4924)	(0, 11, 151, 698, 2541, 6413, 10889, 12338, 8186, 2463)
	$12^{\bullet}$	1.9257	-	-	2220210302	2220230102	-	(0, 11, 151, 700, 2536, 6417, 10891, 12333, 8189, 2462)
	$16^{\diamond}$	1.7553	1.5750	1.6081	3222002102	1222002302	(0, 17, 231, 1050, 3759, 9712, 16269, 18519, 12285, 3692)	(0, 6.5, 116.4, 528.9, 1868.9, 4872.4, 8127.1, 9251.6, 6151.6, 1843.6)
	$16^{\bullet}$	1.7583	-	-	1030120222	3010320222	-	(0, 6.2, 115.5, 532.3, 1874.5, 4846.1, 8156.5, 9241.3, 6149.5, 1845.2)
	$20^{\star}$	1.6629	1.4537	1.5013	2012011022	2032033022	(0, 11, 188, 848, 2988, 7776, 13035, 14790, 9840, 2952)	(0, 4, 93.4, 428, 1492.7, 3878.2, 6540.6, 7372.6, 4929.4, 1474.6)
	$24^{\star}$	1.6059	1.3764	1.4335	0313032021	0131012023	(0, 8, 157, 714, 2481, 6476, 10880, 12309, 8206, 2459)	(0, 2.7, 77.5, 360.7, 1239.1, 3234.8, 5446.7, 6141.8, 4116.7, 1224.3)
	$28^{\diamond}$	1.5701	1.3202	1.3816	0012312110	0032132330	(0, 5, 138, 616, 2110, 5566, 9324, 10545, 7038, 2107)	(0, 1.9, 66.8, 309.1, 1056, 2785, 4662.8, 5266.8, 3521.8, 1053.3)
	$28^{\bullet}$	1.5745	-	-	0113102100	0331302300	-	(0, 1.7, 68.1, 306.9, 1057.8, 2782.5, 4662.5, 5275.2, 3512.4, 1056.4)
	$32^{\diamond}$	1.5443	1.2852	1.3484	0023010231	0021030213	(0, 3.3, 120.3, 541.9, 1851.8, 4844.7, 8193.7, 9202.9, 6166.2, 1842.3)	(0, 1.3, 58.1, 271.9, 932.9, 2413.4, 4089.9, 4621.4, 3071.1, 923)
	$32^{\bullet}$	1.5457	-	-	320012330	120032110	-	(0, 1.2, 58.8, 274.6, 922.5, 2424.1, 4088.6, 4614, 3079.1, 920.2)
	$36^{\star}$	1.5242	1.2978	1.3586	2222222222	2222222222	(0, 2.8, 104.3, 487.7, 1645.4, 4295.4, 7296.8, 8173.8, 5482.3, 1637.6)	(0, 1, 50.7, 247.5, 822.1, 2144.6. 3642.1, 4093.8, 2747.9, 812.8)

* U-uniform minimum aberration combined design.

$^{\diamond}$ U-uniform nearly minimum aberration combined design.

$^{\bullet}$ U-minimum aberration nearly uniform combined design.

4.5 Key Findings for Folding Over High-Level Designs

[66] extended the work of [110] to asymmetric designs with a mixture of two-level and q-level factors, i.e., $\mathbf{d} \in \mathbb{U}_n(2^{s_1} q^{s_2})$. New analytical expressions and lower bounds for the wrap-around discrepancy (WDisc) of combined designs were derived, providing useful benchmarks for searching optimal fold-over plans and optimal combined designs. The equivalence between any fold-over plan and its complementary fold-over plan was established, reducing the search space by half. The uniformity relationships among initial, fold-over, and combined designs were also examined. To illustrate the practical utility of these results, the authors applied the method to the router-bit-life experiment, showing that the new design outperforms the one used in [112] (see Table 18, source [66]). It is worth noting that when $s_1 = 0$ the results specialize to symmetric designs with any number of levels, $\mathbf{d} \in \mathbb{U}_n(q^{s_2})$. The main findings confirm that fold-over plans and their complementary plans are equivalent under the WDisc, thereby halving the search space. Furthermore, Figure 13 (source [66]) demonstrates that the lower bound from [66] is sharper than the one in [113].

[67] extended the work of [66] to asymmetric designs with a mixture of q_1-level and q_2-level factors, $\mathbf{d} \in \mathbb{U}_n(q_1^{s_1} q_2^{s_2})$. Through theoretical analysis, the authors proved that the initial design is uniform if and only if the fold-over design is uniform, the combined design is uniform if and only if its complementary combined design is uniform, and the combined design is uniform (or a good approximation) if and only if the initial design is uniform. Their results also show that fold-over plans and their complementary plans are equivalent under the WDisc and LDisc, again reducing the search space by half. [68] and [69] further extended the work to the most general case involving any number of factors with any number of distinct levels, $\mathbf{d} \in \mathbb{U}_n(q_1^{s_1} q_2^{s_2} \ldots q_m^{s_m})$. They proposed novel analytical expressions and lower bounds for the WDisc and LDisc of combined designs, which can be used to evaluate the optimality of fold-over plans. The investigation of equivalent fold-over designs provides a means to reduce computational complexity. The main results indicate that fold-over plans and their complementary plans are equivalent under the WDisc and LDisc, thereby halving the search space. Moreover, the results highlight that two-stage sequential experimental designs achieve greater precision than single-stage designs of the same size. Using these results as benchmarks, Algorithm 3—a general version of Algorithm 2—can be implemented to search for optimal fold-over plans for any design with arbitrary numbers of factors, levels, and runs.

Table 18: Designing router bit initial experimental design. Source [66]

Factors	Levels	$\mathbf{d}^\top$
A. Suction (in. Hg)	1 2	- - - - - - - - + + + + + + + + - - - - - - - - + + + + + + + +
B. $x - y$ feed (in.lmin.)	60 80	- - - - + + + + - - - - + + + + - - - - + + + + - - - - + + + +
C. In-feed (in.lmin.)	10 50	- - - - + + + + + + + + - - - - - - - - + + + + + + + + - - - -
D. Suction foot	SR BB	- + - + + - + - - + - + + - + - - + - + + - + - - + - + + - + -
E. Stacking height (in.)	3/16 1/4	- + + - + - - + + - - + - + + - - + + - + - - + + - - + - + + -
F. Slot depth (mils)	60 100	- - + + - - + + + + - - + + - - - - + + - - + + + + - - + + - -
G. Speed (rpm)	30,000 40,000	- - - - - - - - - - - - - - - - + + + + + + + + + + + + + + + +
H. Bit type	1 2 3 4	1 2 3 4 3 4 1 2 4 3 2 1 2 1 4 3 1 2 3 4 3 4 1 2 4 3 2 1 2 1 4 3
J. Spindle position	1 2 3 4	1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3

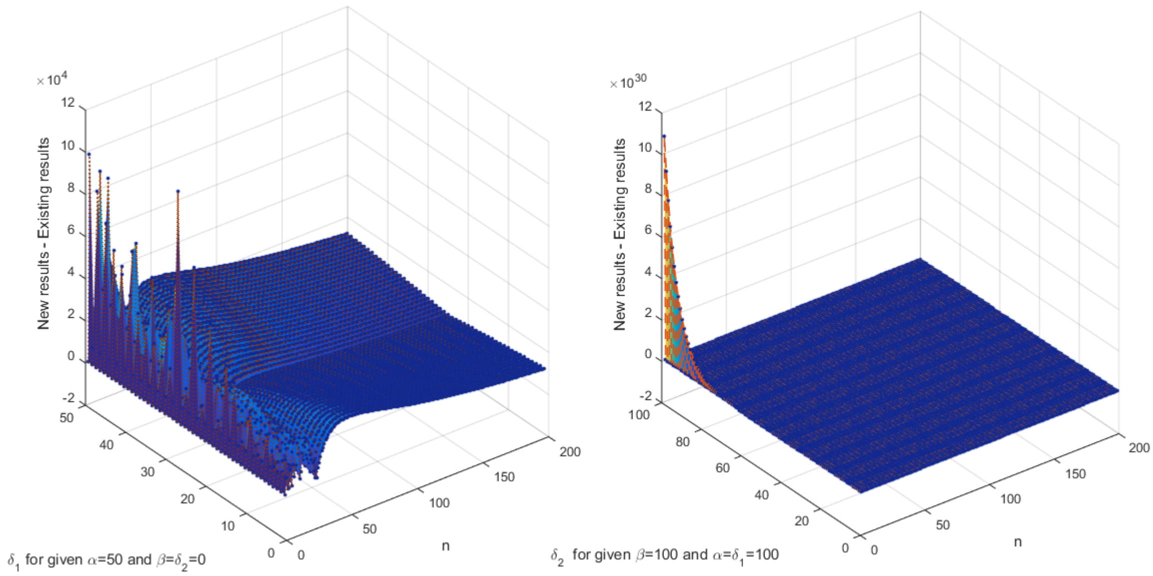


Figure 13: Difference between the lower bound in [66] and the lower bound in [113]. Source [66]

Algorithm 3. Finding optimal fold-over plans for any design with any number of factors, levels and runs $\mathbf{d} \in \mathbb{U}_n(q_1^{s_1} q_2^{s_2} ... q_m^{s_m}).Source[68]$

Input the original balanced design $\mathbf{d} \in \mathbb{U}_n(q_1^{s_1} q_2^{s_2} ... q_m^{s_m})$;
Generate the set of all the fold-over plans Γ;
Define the set of all the complementary fold-over plans Γ^c;
Define the effective searching space of the fold-over plans $\overline{\Gamma} = \Gamma/\Gamma^c$;
Set the current foldover plan $\Gamma^* = \Gamma_0 = (0, \dots, 0)$, where Γ_0 is the initial foldover plan;
Generate the fold-over design $\mathbf{F}^*$ and combined design $\mathbf{C}^* = \begin{pmatrix} \mathbf{d} \\ \mathbf{F}^* \end{pmatrix}$ for the current fold-over plan Γ^*;
Set $Disc^* = [Disc(\mathbf{C}^*)]^2$;
for $k = 1$ to $|\overline{\Gamma}|$ **do**
    Compute $[Disc(\mathbf{C}_k)]^2$, where $\mathbf{C}_k$ is the combined design via the k-th fold-over plan Γ_k from the set of effective fold-over plans $\overline{\Gamma}$;
    if $[Disc(\mathbf{C}_k)]^2 = LBDisc$, where $LBDisc$ is the new lower bound in [69];
        Replace Γ^* by Γ_k;
        Replace $Disc^*$ by $[Disc(\mathbf{C}_k)]^2$;
        break the current **for** loop
    elseif $[Disc(\mathbf{C}_k)]^2 < Disc^*$
        Replace Γ^* by Γ_k;
        Replace $Disc^*$ by $[Disc(\mathbf{C}_k)]^2$;
    end if
end for
end for
Output Γ^* and $[Disc(\mathbf{C}^*)]^2$.

4.6 Recommending an Efficient Fold-Over Technique

[85] provides an in-depth examination of fold-over techniques and presents a comparative study of widely used methods, aiming to help experimenters select a suitable technique for their experiments. The work addresses two fundamental questions: (1) Do these techniques significantly reduce the confounding present in the initial designs? (2) Do the resulting combined designs (obtained by joining the initial design with its fold-over) differ substantially in optimality despite the markedly different search domains associated with each technique? The optimality criteria considered are aberration, confounding, Hamming distance, and uniformity. [85] establishes a general framework that connects initial designs and combined designs for any fold-over technique. To answer these questions, numerical results, comparisons, and discussions based on the above criteria are provided for a variety of orthogonal designs and uniform designs. To illustrate the practical use of the theoretical results, the paper includes a catalog of optimal fold-over plans for constructing uniform minimum-aberration symmetric three-level and four-level combined designs, suitable for experiments with either qualitative or quantitative factors.

Tables 19-21 (source [85]) list the optimal fold-over plans obtained via the WDisc, HDP, and GWLP criteria, along with their corresponding dictionary cross-entropy loss function (DCELF) values for three- and four-level uniform designs. Table 22 (source [85]) reports the computational time required to find the optimal fold-over plans. While it is often assumed that a larger search domain improves the chances of obtaining an optimal combined design, the distinctions in computational complexity between different fold-over methods—and whether they lead to widely differing optimal combined designs—are seldom questioned. From Tables 19-22, it can be seen that differences in computational time become more pronounced as the number of factors and levels increases. For small designs, the search time varies by less than a minute, whereas for larger designs the difference can extend to several days. For example, obtaining the optimal combined design C^* for $D(48, 3, 9)$ required 63.8651s for T_A, 2866.2066 s for T_R, and 454928.6641 s for T_P.

The differences in DCELF values among the three methods are minor for uniform designs. For instance, the WDisc value of C^* for $D(39, 3, 9)$ decreases to 1.789987, 1.790290, and 1.788280 for T_A, T_R, and T_P, respectively. The DCELF(HDP) values for C^* in $D(42, 3, 8)$ are 7.026917, 7.010097, and 7.005712, and the DCELF(GWLP) values for C^* in $D(21, 3, 7)$ are 6.021137, 6.010674, and 6.008198—all very close to one another. No obvious distinction emerges because the overall pairwise differences in WDisc, DCELF(HDP), and DCELF(GWLP) are confined to the narrow intervals [0,0.0772], [0, 1.8113], and [0, 0.4015], respectively. For designs of the form $D(n, 4, 2)$, the WDisc difference does not exceed 4×10^{-5}, and the DCELF(HDP) and DCELF(GWLP) measures are identical. Although some differences are noticeable, most lie within the interval [0,1] and are very close to zero, indicating that the choice of fold-over method does not lead to substantially different optimal combined designs. Consequently, constructing a larger search domain solely to locate the optimal combined design is inefficient, uneconomical, and generally inadvisable.

Table 19: Optimal foldover plans for the uniform designs $\mathbf{d} \in \mathbb{U}_n(3^s)$ for $2 \le s \le 5$. Source [85]

Design	WDisc				DCELF(HDP)				DCELF(GWLP)			
$D(n,q,s)$	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P
$D(9,3,2)$	0.049726	0.049726*	0.049726*	0.049726*	2.384923	3.042716*	3.042716*	3.042716*	0.000000	0.000000*	0.000000*	0.000000*
$D(12,3,2)$	0.050412	0.049897*	0.049897*	0.049897*	3.011117	3.118071*	3.118071*	3.118071*	1.002808	1.000177*	1.000177*	1.000177*
$D(15,3,2)$	0.050165	0.049945	0.049835*	0.049835*	3.027865	3.205400	3.197080*	3.197080*	1.001153	1.000288	1.000072*	1.000072*
$D(21,3,2)$	0.049950	0.049782*	0.049782*	0.049782*	3.082670	3.354385*	3.354385*	3.354385*	1.000300	1.000019*	1.000019*	1.000019*
$D(24,3,2)$	0.049897	0.049811	0.049769*	0.049769*	3.118071	3.426397	3.422419*	3.422419*	1.000176	1.000044	1.000011*	1.000011*
$D(27,3,2)$	0.049726	0.049726*	0.049726*	0.049726*	3.147695	3.478401*	3.478401*	3.478401*	0.000000	0.000000*	0.000000*	0.000000*
$D(30,3,2)$	0.049835	0.049753*	0.049753*	0.049753*	3.197080	3.532047*	3.532047*	3.532047*	1.000072	1.000005*	1.000005*	1.000005*
$D(33,3,2)$	0.049816	0.049748*	0.049748*	0.049748*	3.237901	3.575166*	3.575166*	3.575166*	1.000049	1.000003*	1.000003*	1.000003*
$D(36,3,2)$	0.049726	0.049726*	0.049726*	0.049726*	3.271455	3.610537*	3.610537*	3.610537*	0.000000	0.000000*	0.000000*	0.000000*
$D(39,3,2)$	0.049791	0.049742*	0.049742*	0.049742*	3.317183	3.643486*	3.643486*	3.643486*	1.000025	1.000002*	1.000002*	1.000002*
$D(42,3,2)$	0.049782	0.049740*	0.049740*	0.049740*	3.354385	3.670636*	3.670636*	3.670636*	1.000019	1.000001*	1.000001*	1.000001*
$D(45,3,2)$	0.049726	0.049726*	0.049726*	0.049726*	3.384923	3.693440*	3.693440*	3.693440*	0.000000	0.000000*	0.000000*	0.000000*
$D(48,3,2)$	0.049769	0.049747	0.049736*	0.049736*	3.422419	3.715324	3.714724*	3.714724*	1.000011	1.000003	1.000001*	1.000001*
$D(51,3,2)$	0.049764	0.049735*	0.049735*	0.049735*	3.453047	3.732776*	3.732776*	3.732776*	1.000009	1.000001*	1.000001*	1.000001*
$D(9,3,3)$	0.100956	0.100550	0.100347*	0.100347*	2.552400	4.004981	3.271455*	3.271455*	1.384923	1.147695	1.042716*	1.042716*
$D(12,3,3)$	0.103281	0.101480	0.100966	0.100852*	3.090204	4.004981	4.002808	3.498605*	2.024593	2.004382	2.001584*	2.001584*
$D(15,3,3)$	0.102151	0.100950	0.100670	0.100645*	3.197080	4.007150	4.007150	4.003192*	2.010256	2.001799	2.000649*	2.000649*
$D(18,3,3)$	0.100347	0.100194*	0.100194*	0.100194*	3.271455	4.011117*	4.011117*	4.011117*	1.042716	1.002808*	1.002808*	1.002808*
$D(21,3,3)$	0.101118	0.100555	0.100400*	0.100400*	3.413768	4.022364*	4.022364*	4.022364*	2.002693	2.000469	2.000169*	2.000169*
$D(24,3,3)$	0.100852	0.100440	0.100330*	0.100330*	3.498605	4.033112*	4.033112*	4.033112*	2.001584	2.000276	2.000099*	2.000099*
$D(27,3,3)$	0.100144	0.100144*	0.100144*	0.100144*	3.552400	4.042716*	4.042716*	4.042716*	0.000000	0.000000*	0.000000*	0.000000*
$D(30,3,3)$	0.100597	0.100333	0.100263*	0.100263*	4.001799	4.069670*	4.069670*	4.069670*	2.000649	2.000113	2.000041*	2.000041*
$D(33,3,3)$	0.100538	0.100310	0.100247*	0.100247*	4.005917	4.095101*	4.095101*	4.095101*	2.000443	2.000077	2.000028*	2.000028*
$D(36,3,3)$	0.100194	0.100156*	0.100156*	0.100156*	4.011117	4.118071*	4.118071*	4.118071*	1.002808	1.000177*	1.000177*	1.000177*
$D(39,3,3)$	0.100441	0.100236	0.100221	0.100218*	4.016723	4.160200	4.147695	4.141519*	2.000227	2.000014*	2.000014*	2.000014*
$D(42,3,3)$	0.100400	0.100253	0.100229	0.100201*	4.022364	4.179885	4.173956	4.165139*	2.000169	2.000030	2.000011*	2.000011*
$D(45,3,3)$	0.100176	0.100160	0.100152*	0.100152*	4.027865	4.205400	4.197080*	4.197080*	1.001153	1.000288	1.000072*	1.000072*
$D(48,3,3)$	0.100330	0.100227	0.100193*	0.100193*	4.033112	4.225183*	4.225183*	4.225183*	2.000099	2.000017	2.000006*	2.000006*
$D(51,3,3)$	0.100300	0.100209	0.100185*	0.100185*	4.038068	4.249836*	4.249836*	4.249836*	2.000078	2.000014	2.000005*	2.000005*
$D(9,3,4)$	0.183671	0.181198*	0.181198*	0.181198*	2.656543	4.042716*	4.042716*	4.042716*	2.826728	2.610537*	2.610537*	2.610537*
$D(12,3,4)$	0.188764	0.182211	0.182065	0.181870*	3.271455	4.064832	4.072972	4.024593*	3.090204	3.006291*	3.006291*	3.006291*
$D(15,3,4)$	0.185369	0.182206	0.181170	0.180910*	3.461356	5.000200	4.172199	4.093799*	3.039530	3.010256	3.002589*	3.002589*
$D(18,3,4)$	0.180419	0.179767	0.179584*	0.179584*	3.552400	4.209574*	4.209574*	4.209574*	2.384923	2.118071	2.042716*	2.042716*
$D(21,3,4)$	0.182148	0.180412	0.180202*	0.180202*	3.678581	5.000408	4.265654*	4.265654*	3.010674	3.001517	3.000675*	3.000675*
$D(24,3,4)$	0.181286	0.180054	0.180039*	0.180039*	3.736469	5.003816	4.330438*	4.330438*	3.006291	3.000890*	3.000890*	3.000890*
$D(27,3,4)$	0.179335	0.179290*	0.179290*	0.179290*	3.769141	4.384923*	4.384923*	4.384923*	1.384923	1.042716*	1.042716*	1.042716*
$D(30,3,4)$	0.180567	0.179918	0.179765	0.179761*	4.027865	5.001799	5.000451	4.494660*	3.002589	3.000649	3.000365*	3.000365*
$D(33,3,4)$	0.180448	0.179831	0.179645*	0.179645*	4.085389	5.000662	5.001487	5.000373*	3.001771	3.000443	3.000111*	3.000111*
$D(36,3,4)$	0.179572	0.179355	0.179352*	0.179352*	4.147695	5.003464	5.002808*	5.002808*	2.042716	2.002808*	2.002808*	2.002808*
$D(39,3,4)$	0.180189	0.179718	0.179546	0.179514*	4.204775	5.005769	5.004243	5.001065*	3.000908	3.000227	3.000057*	3.000057*
$D(42,3,4)$	0.180063	0.179674	0.179536	0.179481*	4.253978	5.004981	5.005712	5.002063*	3.000675	3.000169	3.000042*	3.000042*
$D(45,3,4)$	0.179465	0.179326	0.179324	0.179323*	4.295511	5.006384	5.007150	5.003192*	2.018056	2.001153*	2.001153*	2.001153*
$D(48,3,4)$	0.179829	0.179501	0.179500	0.179498*	4.330438	5.007002	5.008543	5.006291*	3.000396	3.000056*	3.000056*	3.000056*
$D(51,3,4)$	0.179722	0.179498	0.179445	0.179443*	4.359912	5.015304	5.009863	5.007574*	3.000310	3.000078	3.000044*	3.000044*
$D(9,3,5)$	0.338644	0.320505	0.312061	0.309849*	3.147695	5.004981	4.084319*	4.084319*	4.384923	4.147695	4.042716	4.013683*
$D(12,3,5)$	0.324634	0.310790	0.308754	0.308213*	3.478401	5.002808	5.001249	5.000313*	4.209574	4.047907	4.017237*	4.017237*
$D(15,3,5)$	0.316546	0.308242	0.306578	0.305408*	4.003192	5.019551	5.007150	5.001799*	4.101094	4.025232	4.012011	4.007150*
$D(18,3,5)$	0.304900	0.302534	0.302256	0.302038*	3.723484	5.030152	5.011117*	5.011117*	3.723484	3.384923	3.301435	3.209574*
$D(21,3,5)$	0.308997	0.304557	0.304394	0.303974*	4.054906	5.042716	5.022364*	5.022364*	4.028978	4.007443	4.004781*	4.004781*
$D(24,3,5)$	0.307071	0.303638	0.303199*	0.303199*	4.118071	5.036192	5.033112*	5.033112*	4.017237	4.004376	4.002808*	4.002808*
$D(27,3,5)$	0.302611	0.301576	0.301608	0.301374*	4.147695	5.076692	5.062198	5.042716*	3.384923	3.096213	3.108544	3.042716*
$D(30,3,5)$	0.304721	0.302468	0.302095	0.302027*	4.287562	5.090204	5.079713	5.069670*	4.007150	4.000883	4.000451*	4.000451*
$D(33,3,5)$	0.303835	0.302178	0.302165	0.302156*	4.345789	6.000042	5.105147*	5.105147*	4.004898	4.000787*	4.000787*	4.000787*
$D(36,3,5)$	0.302024	0.301291	0.301250	0.301243*	5.000139	5.121292	5.111696	5.087245*	3.230366	3.036192	3.027302	3.024593*
$D(39,3,5)$	0.303240	0.302012	0.301712	0.301575*	4.562275	6.000267	5.176013	5.117333*	4.002516	4.000569	4.000158*	4.000158*
$D(42,3,5)$	0.302915	0.301645	0.301568	0.301496*	4.617770	6.000026	5.200652	5.156381*	4.001872	4.000230	4.000117*	4.000117*
$D(45,3,5)$	0.301623	0.301181	0.301125	0.301085*	5.001423	6.000355	5.233129	5.210974*	3.113087	3.021903	3.011408	3.008636*
$D(48,3,5)$	0.302334	0.301465	0.301352*	0.301352*	5.000704	6.000489	5.248478	5.238149*	4.001098	4.000134	4.000069*	4.000069*

[a] The uniform designs are gathered from http://www.math.hkbu.edu.hk/UniformDesign/.

[b] * refers to the best combined design among the optimal combined designs obtained from different fold-over methods.

Table 20: Optimal foldover plans for the uniform designs $\mathbf{d} \in \mathbb{U}_n(3^s)$ for $6 \leq s \leq 9$. Source [85]

Design	WDisc				DCELF(HDP)				DCELF(GWLP)			
$D(n,q,s)$	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P
$D(9,3,6)$	0.583565	0.519519	0.515121	0.510153*	3.384923	5.008807	5.002220	4.298146*	5.656543	5.188770	5.147695	5.072972*
$D(12,3,6)$	0.537986	0.510047	0.507147	0.502276*	3.634848	6.000313	5.042716	5.002808*	5.358301	5.125328	5.090204	5.037791*
$D(15,3,6)$	0.521043	0.501774	0.498437	0.496548*	4.048247	6.000200	5.072972	5.048247*	5.197080	5.056456	5.022718	5.015913*
$D(18,3,6)$	0.497194	0.490172	0.489223	0.489090*	4.034120	6.002220	5.090204*	5.090204*	4.877680	4.711683	4.642349	4.623044*
$D(21,3,6)$	0.503369	0.493696	0.492999	0.491847*	4.182843	5.288688	5.165139*	5.165139*	5.062405	5.015489	5.010674*	5.010674*
$D(24,3,6)$	0.499805	0.491488	0.490178	0.489691*	4.418406	6.002808	5.245882	5.173238*	5.037791	5.007371	5.002470*	5.002470*
$D(27,3,6)$	0.490921	0.487608	0.486516	0.486283*	4.505080	7.000062	5.188770*	5.188770*	4.740537	4.449537	4.230366	4.147695*
$D(30,3,6)$	0.494298	0.488930	0.488573	0.487919*	4.617301	6.006014	6.001799	6.000802*	5.015913	5.003774	5.001983	5.001013*
$D(33,3,6)$	0.492588	0.488368	0.487756	0.487631*	5.001487	6.006939	6.001035	6.000802*	5.010933	5.002580	5.000997*	5.000997*
$D(36,3,6)$	0.488524	0.486326	0.486267	0.486125*	5.016500	6.008807	6.002808	6.002220*	4.568267	4.204375	4.186179	4.155306*
$D(39,3,6)$	0.490501	0.487283	0.487079	0.486656*	5.009494	6.006614*	6.009494	6.006614*	5.005642	5.001065	5.000695	5.000355*
$D(42,3,6)$	0.489724	0.487117	0.486716	0.486276*	5.025361	6.016915	6.011117	6.008198*	5.004205	5.000986	5.000380	5.000264*
$D(45,3,6)$	0.487291	0.485935	0.485784	0.485700*	5.077443	7.000022	6.012628	6.007958*	4.390443	4.126994	4.093799	4.079713*
$D(48,3,6)$	0.488535	0.486537	0.486208	0.486162*	5.077159	7.000020	6.015060	6.013013*	5.002470	5.000465	5.000223*	5.000223*
$D(9,3,7)$	0.967084	0.839948	0.836656	0.818396*	3.552400	6.002220	5.042716	5.008807*	6.769141	6.396350	6.384923	6.188770*
$D(12,3,7)$	0.871290	0.808466	0.804506	0.802074*	4.011117	6.007750	5.209574	5.137661*	6.498605	6.155306	6.132708	6.090204*
$D(15,3,7)$	0.836729	0.794236	0.789798	0.788287*	4.197080	6.019551	6.000802	5.279526*	6.314328	6.083859	6.049393*	6.049393*
$D(18,3,7)$	0.791951	0.774775	0.772349	0.771564*	4.271455	6.022854	5.349140*	5.349140*	5.936987	5.866382	5.836914	5.821143*
$D(21,3,7)$	0.803240	0.779708	0.776734	0.775242*	5.000408	6.025361	6.006492	6.004981*	6.113956	6.021137	6.010674	6.008198*
$D(24,3,7)$	0.794090	0.776550	0.773568	0.771993*	5.001249	7.000078	6.015060	6.011117*	6.070890	6.021699	6.007943	6.006291*
$D(27,3,7)$	0.777541	0.767439	0.766439	0.766221*	5.034120	7.000247	6.026389	6.008807*	5.883758	5.732276	5.688451	5.675287*
$D(30,3,7)$	0.782464	0.770070	0.768825	0.767793*	5.023547	6.072972	6.116158	6.032504*	6.030614	6.006800	6.003273	6.002589*
$D(33,3,7)$	0.778597	0.767363	0.767262	0.766516*	5.067022	7.000166	6.126008	6.058444*	6.021161	6.002763	6.002240	6.001771*
$D(36,3,7)$	0.771216	0.764744	0.763861	0.763620*	6.000139	7.000868	7.000035	6.072972*	5.817216	5.593831	5.511412	5.478401*
$D(39,3,7)$	0.773395	0.766099	0.765316	0.764768*	5.179190	7.000474	7.000030	6.129301*	6.010986	6.002157	6.001150	6.000695*
$D(42,3,7)$	0.771920	0.765300	0.764440	0.764227*	5.259828	7.000638	6.197679	6.165139*	6.008198	6.001781	6.000675*	6.000675*
$D(45,3,7)$	0.767361	0.763069	0.762671	0.762473*	6.000089	7.001423	7.000200	6.188770*	5.719808	5.432388	5.370876	5.335722*
$D(48,3,7)$	0.768411	0.763635	0.763044	0.762825*	6.000078	7.000957	7.000020*	7.000020*	6.004827	6.000942	6.000501	6.000303*
$D(9,3,8)$	1.563799	1.332985	1.298661	1.291631*	3.656543	6.019551	5.147695	5.062198*	7.826728	7.552400	7.384923	7.349140*
$D(12,3,8)$	1.407766	1.274955	1.259385	1.259254*	4.147695	6.042716	6.011117	6.001249*	7.675973	7.351437	7.232980*	7.232980*
$D(15,3,8)$	1.325360	1.236861	1.232756	1.228319*	4.449537	7.000802	6.027865	6.015913*	7.461356	7.138067	7.123878	7.096693*
$D(18,3,8)$	1.251733	1.202133	1.199946	1.197502*	4.552400	6.114872	6.034120*	6.034120*	7.042716	7.004981	7.002808	7.000868*
$D(21,3,8)$	1.256325	1.209292	1.203931	1.200491*	5.031831	7.004981	6.182843	6.072972*	7.182843	7.042716	7.028978	7.017650*
$D(24,3,8)$	1.238675	1.199445	1.196509	1.194236*	5.108544	7.002808	7.000078	6.152746*	7.118071	7.028718	7.017237	7.010443*
$D(27,3,8)$	1.210095	1.185241	1.184035	1.182455*	5.193384	7.007415	7.000247	6.125610*	7.001757	6.856867	7.000110	6.825109*
$D(30,3,8)$	1.215533	1.188905	1.186741	1.184912*	5.287562	7.004981	7.001249	6.242787*	7.052878	7.013413	7.006125	7.005179*
$D(33,3,8)$	1.208341	1.184662	1.183703	1.182506*	5.409499	7.008033	7.002025	7.001487*	7.036874	7.006465	7.004194	7.002949*
$D(36,3,8)$	1.194565	1.178351	1.176980	1.176616*	6.002220	7.004981	7.004189	7.001700*	7.000313	7.000078	7.000020	6.726291*
$D(39,3,8)$	1.197453	1.180037	1.179317	1.178135*	6.000118	7.011685	7.006614	7.004981*	7.019335	7.003774	7.002157	7.001514*
$D(42,3,8)$	1.194021	1.179354	1.177686	1.176878*	6.000408	7.026917	7.010097	7.005712*	7.014458	7.003163	7.001353*	7.001353*
$D(45,3,8)$	1.185784	1.174547	1.173864*	1.173864*	6.019551	8.000200	7.010631	7.008807*	7.000228	6.664569	7.000014	6.598550*
$D(48,3,8)$	1.187932	1.175738	1.175217	1.174363*	6.004981	8.000020	7.016132	7.008543*	7.008543	7.001155	7.001098	7.000794*
$D(9,3,9)$	2.559674	2.114787	2.081621	2.037556*	4.147695	7.000557	6.008807	6.000557*	8.884476	8.723484	8.681997	8.579960*
$D(12,3,9)$	2.262547	1.983964	1.967164	1.945461*	4.478401	7.000313	6.081425	6.030152*	8.794694	8.531174	8.467842	8.371772*
$D(15,3,9)$	2.090500	1.918042	1.896015	1.891501*	5.000802	7.007150	7.000802	6.131712*	8.635764	8.323576	8.222064	8.187100*
$D(18,3,9)$	1.965419	1.856700	1.846676	1.842810*	4.748320	8.000139	6.367324	6.237280*	8.271455	8.067506	8.042716	8.026389*
$D(21,3,9)$	1.939092	1.844241	1.837540	1.833997*	6.000408	8.000102	7.003666	7.000408*	8.263992	8.060496	8.041078	8.031831*
$D(24,3,9)$	1.905719	1.830939	1.823481	1.820994*	5.358301	8.000078	7.011117	7.003816*	8.178395	8.047907	8.024593	8.016405*
$D(27,3,9)$	1.848208	1.800432	1.794306*	1.794306*	5.429034	8.000062	7.000989	7.000557*	7.954466	7.898150	7.873333*	7.873333*
$D(30,3,9)$	1.863337	1.808523	1.805304	1.802634*	6.004981	8.000050	7.066424	7.021507*	8.083859	8.020326	8.011560	8.007876*
$D(33,3,9)$	1.849349	1.802259	1.798889	1.796573*	6.023162	8.001035	7.045206	7.025078*	8.059199	8.010579	8.007030	8.005399*
$D(36,3,9)$	1.826897	1.791494	1.789283	1.786895*	6.062198	8.000557	7.144336	7.034120*	8.003464	8.000263	8.000217	8.000055*
$D(39,3,9)$	1.828311	1.789987	1.790290	1.788280*	6.044809	8.000267	8.000030	7.067914*	8.031460	8.005461	8.004579	8.002777*
$D(42,3,9)$	1.821110	1.788467	1.786875	1.785237*	6.082670	8.001249	8.000026	7.095341*	8.024924	8.004927	8.003163	8.002808*
$D(45,3,9)$	1.807815	1.782510	1.780239	1.779659*	6.230366	8.003192	8.000089	7.134347*	8.000512	8.000107	8.000032	7.778213*
$D(48,3,9)$	1.809151	1.783881	1.782012	1.780764*	6.168072	8.002808	8.000489	8.000078*	8.014014	8.003748	8.001391	8.001211*

[a] The uniform designs are gathered from http://www.math.hkbu.edu.hk/UniformDesign/.
[b] * refers to the best combined design among the optimal combined designs obtained from different fold-over methods.

Table 21: Optimal foldover plans for the uniform designs $\mathbf{d} \in \mathbb{U}_n(4^s)$ for $2 \le s \le 5$. Source [85]

Design	WDisc				DCELF(HDP)				DCELF(GWLP)			
$D(n,q,s)$	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P	Initial	T_A	T_R	T_P
$D(8,4,2)$	0.028863	0.027886*	0.027886*	0.027886*	2.147695	2.552400*	2.552400*	2.552400*	1.147695	0.000000*	0.000000*	0.000000*
$D(12,4,2)$	0.028754	0.028103*	0.028103*	0.028103*	2.384923	3.019551*	3.019551*	3.019551*	1.019551	1.002220*	1.002220*	1.002220*
$D(16,4,2)$	0.027886	0.027886*	0.027886*	0.027886*	2.552400	3.042716*	3.042716*	3.042716*	0.000000	0.000000*	0.000000*	0.000000*
$D(20,4,2)$	0.028199	0.027964*	0.027964*	0.027964*	3.007150	3.101094*	3.101094*	3.101094*	1.002589	1.000288*	1.000288*	1.000288*
$D(24,4,2)$	0.027995	0.027886*	0.027886*	0.027886*	3.019551	3.147695*	3.147695*	3.147695*	1.002220	0.000000*	0.000000*	0.000000*
$D(28,4,2)$	0.028046	0.027966	0.027926*	0.027926*	3.031831	3.218494*	3.218494*	3.218494*	1.000675	1.000075*	1.000075*	1.000075*
$D(32,4,2)$	0.027886	0.027886*	0.027886*	0.027886*	3.042716	3.271455*	3.271455*	3.271455*	0.000000	0.000000*	0.000000*	0.000000*
$D(36,4,2)$	0.027983	0.027910*	0.027910*	0.027910*	3.072972	3.336737*	3.336737*	3.336737*	1.000247	1.000027*	1.000027*	1.000027*
$D(40,4,2)$	0.027925	0.027886*	0.027886*	0.027886*	3.101094	3.384923*	3.384923*	3.384923*	1.000288	0.000000*	0.000000*	0.000000*
$D(44,4,2)$	0.027951	0.027919	0.027902*	0.027902*	3.126008	3.438481*	3.438481*	3.438481*	1.000111	1.000012*	1.000012*	1.000012*
$D(48,4,2)$	0.027886	0.027886*	0.027886*	0.027886*	3.147695	3.478401*	3.478401*	3.478401*	0.000000	0.000000*	0.000000*	0.000000*
$D(52,4,2)$	0.027933	0.027909	0.027898*	0.027898*	3.185556	3.520454*	3.520454*	3.520454*	1.000057	1.000006*	1.000006*	1.000006*
$D(8,4,3)$	0.060904	0.057791*	0.057791*	0.057791*	0.055991	3.042716*	3.042716*	3.042716*	2.552400	2.147695*	2.147695*	2.147695*
$D(12,4,3)$	0.059914	0.057337	0.057337	0.057055*	2.552400	3.147695*	3.147695*	3.147695*	2.147695	2.019551*	2.019551*	2.019551*
$D(16,4,3)$	0.056357	0.056113*	0.056113*	0.056113*	2.693440	3.271455*	3.271455*	3.271455*	1.552400	1.147695*	1.147695*	1.147695*
$D(20,4,3)$	0.057432	0.056632	0.056510	0.056397*	3.060124	3.478401	3.461356*	3.461356*	2.022718	2.002589*	2.002589*	2.002589*
$D(24,4,3)$	0.056591	0.056191	0.056191	0.056164*	3.147695	4.001249	4.001249	3.573350*	2.019551	2.002220*	2.002220*	2.002220*
$D(28,4,3)$	0.056681	0.056248	0.056248	0.056164*	3.218494	3.650593	3.647546	3.644441*	2.006040	2.000675*	2.000675*	2.000675*
$D(32,4,3)$	0.056021	0.056021	0.056021	0.055991*	3.271455	3.693440*	3.693440*	3.693440*	1.147695	0.000000*	0.000000*	0.000000*
$D(36,4,3)$	0.056409	0.056146	0.056146	0.056095*	3.384923	4.002220*	4.002220*	4.002220*	2.002220	2.000247*	2.000247*	2.000247*
$D(40,4,3)$	0.056207	0.056063	0.056058	0.056053*	3.461356	4.007150*	4.009028	4.007150*	2.002589	2.000288*	2.000288*	2.000288*
$D(44,4,3)$	0.056289	0.056123	0.056098	0.056075*	3.514261	4.013193*	4.013193*	4.013193*	2.000997	2.000111*	2.000111*	2.000111*
$D(48,4,3)$	0.056031	0.055997*	0.055997*	0.055997*	4.001249	4.019551*	4.019551*	4.019551*	1.052103	1.002220*	1.002220*	1.002220*
$D(52,4,3)$	0.056200	0.056058	0.056062	0.056047*	3.606535	4.025827*	4.025827*	4.025827*	2.000512	2.000057*	2.000057*	2.000057*
$D(8,4,4)$	0.126358	0.105292	0.104088*	0.104088*	2.384923	3.042716*	3.042716*	3.042716*	3.769141	3.147695*	3.147695*	3.147695*
$D(12,4,4)$	0.111869	0.103878	0.104806	0.103478*	2.656543	4.001249	4.001249	3.384923*	3.384923	3.072972*	3.072972*	3.072972*
$D(16,4,4)$	0.101929	0.100616*	0.100616*	0.100616*	2.769141	3.552400*	3.552400*	3.552400*	2.884476	2.656543*	2.656543*	2.656543*
$D(20,4,4)$	0.104536	0.101806	0.101736	0.101395*	3.197080	4.015913	4.015913	4.011117*	3.083859	3.010256*	3.010256*	3.010256*
$D(24,4,4)$	0.102013	0.100702	0.100636*	0.100636*	3.384923	4.042716	4.019551*	4.019551*	3.072972	3.008807*	3.008807*	3.008807*
$D(28,4,4)$	0.102318	0.100788	0.100854	0.100704*	3.490089	4.048656	4.058150	4.037102*	3.023622	3.002693*	3.002693*	3.002693*
$D(32,4,4)$	0.100451	0.100034*	0.100034*	0.100034*	3.552400	4.042716*	4.042716*	4.042716*	2.656543	2.147695*	2.147695*	2.147695*
$D(36,4,4)$	0.101359	0.100582	0.100499	0.100377*	3.656543	4.140991	4.147695	4.096213*	3.008807	3.000989*	3.000989*	3.000989*
$D(40,4,4)$	0.100752	0.100222	0.100223	0.100195*	4.002808	4.222064	4.222064	4.156821*	3.010256	3.001153*	3.001153*	3.001153*
$D(8,4,5)$	0.235627	0.187793	0.182980	0.181021*	2.478401	3.271455	3.271455	3.147695*	4.861372	4.552400	4.552400	4.384923*
$D(12,4,5)$	0.197754	0.179820	0.177820	0.176369*	3.019551	4.030152	4.011117	4.004981*	4.673914	4.284898	4.174947*	4.174947*
$D(16,4,5)$	0.174958	0.169773*	0.170045	0.169773*	2.815188	3.723484*	3.723484*	3.723484*	3.953792	3.861372*	3.861372*	3.861372*
$D(20,4,5)$	0.179359	0.172459	0.172149	0.171467*	3.384923	4.197080	4.135680	4.106673*	4.197080	4.039530	4.027865*	4.027865*
$D(24,4,5)$	0.173004	0.169688	0.169555	0.169368*	3.592675	4.271455	4.209574	4.127793*	4.174947	4.034120*	4.034120*	4.034120*
$D(28,4,5)$	0.173838	0.170113	0.169828	0.169489*	4.000918	5.000918	4.384923	4.267095*	4.070279	4.010674	4.007443*	4.007443*
$D(32,4,5)$	0.169500	0.167791*	0.167888	0.167791*	3.723484	4.209574*	4.271455	4.209574*	3.861372	3.478401*	3.552400	3.478401*
$D(36,4,5)$	0.171355	0.168811	0.168856	0.168749*	4.019551	5.001249	5.002808	5.000139*	4.024011	4.002741*	4.002741*	4.002741*

[a] The uniform designs are gathered from http://www.math.hkbu.edu.hk/UniformDesign/.
[b] * refers to the best combined design among the optimal combined designs obtained from different fold-over methods.

Table 22: Computational time for the three fold-over techniques. Source [85]

Design	Computational Time			Design	Computational Time			Design	Computational Time			Design	Computational Time		
$D(n,q,s)$	T_A	T_R	T_P	$D(n,q,s)$	T_A	T_R	T_P	$D(n,q,s)$	T_A	T_R	T_P	$D(n,q,s)$	T_A	T_R	T_P
$D(9,3,6)$	0.2564	0.6916	48.2743	$D(9,3,7)$	0.3677	2.7612	1047.7034	$D(9,3,8)$	0.5666	12.1862	24159.4518	$D(9,3,9)$	1.0583	36.2986	4945.2600
$D(12,3,6)$	0.1100	1.0404	71.5984	$D(12,3,7)$	0.2916	4.3917	1237.8062	$D(12,3,8)$	0.7322	17.0991	25033.9220	$D(12,3,9)$	1.2182	49.4633	6668.7105
$D(15,3,6)$	0.1538	1.6121	110.0229	$D(15,3,7)$	0.3978	6.4691	1738.9944	$D(15,3,8)$	1.0270	24.8844	35824.1974	$D(15,3,9)$	1.8665	70.9389	9907.2689
$D(18,3,6)$	0.2392	2.6841	167.6780	$D(18,3,7)$	0.6375	10.0683	2251.4016	$D(18,3,8)$	1.5659	38.0022	42031.9460	$D(18,3,9)$	2.7282	105.3170	14755.9535
$D(21,3,6)$	0.4362	3.9142	257.7949	$D(21,3,7)$	0.9955	15.3506	2841.1547	$D(21,3,8)$	2.2908	59.5762	46533.9251	$D(21,3,9)$	4.0811	157.4890	22140.8146
$D(24,3,6)$	0.5399	6.0191	390.7350	$D(24,3,7)$	1.4264	24.0190	4246.0267	$D(24,3,8)$	3.5445	91.2014	51023.1385	$D(24,3,9)$	6.1986	242.0032	33787.3629
$D(27,3,6)$	0.8238	9.0896	591.8497	$D(27,3,7)$	2.2779	37.3497	5873.3352	$D(27,3,8)$	5.3452	136.4525	60486.2411	$D(27,3,9)$	9.7070	376.4227	54239.9689
$D(30,3,6)$	1.2301	13.6709	882.9833	$D(30,3,7)$	3.0981	53.4574	7465.5123	$D(30,3,8)$	7.4730	195.2544	75586.1779	$D(30,3,9)$	13.6624	538.0645	81012.6567
$D(33,3,6)$	1.7076	19.6612	1321.5005	$D(33,3,7)$	4.2893	73.6066	13003.7192	$D(33,3,8)$	10.6227	267.8463	90242.3203	$D(33,3,9)$	18.0954	790.3323	111166.4384
$D(36,3,6)$	2.2387	25.8662	1738.4702	$D(36,3,7)$	5.8055	98.6063	16870.9275	$D(36,3,8)$	13.9668	361.9885	118467.4299	$D(36,3,9)$	23.8281	1018.1209	154089.7289
$D(39,3,6)$	2.9888	33.5926	2125.4190	$D(39,3,7)$	7.4972	129.1963	21099.7358	$D(39,3,8)$	18.5446	449.7583	143223.5494	$D(39,3,9)$	31.2544	1255.3264	207341.6545
$D(42,3,6)$	3.7434	42.7345	2679.5429	$D(42,3,7)$	9.7828	141.9781	24600.6970	$D(42,3,8)$	23.9148	471.5839	174463.0042	$D(42,3,9)$	40.3487	1597.7009	283200.0106
$D(45,3,6)$	4.7634	53.8094	3429.4733	$D(45,3,7)$	12.2473	215.3358	29799.8016	$D(45,3,8)$	29.8741	596.6521	222735.2293	$D(45,3,9)$	51.1087	2256.8693	357409.9084
$D(48,3,6)$	5.9432	67.0617	4578.2656	$D(48,3,7)$	15.0961	266.3546	31582.2016	$D(48,3,8)$	37.4222	751.2802	233037.1843	$D(48,3,9)$	63.8651	2866.2066	454928.6641

[a] Time is recorded in second, and it refers to the sum of searching the optimal fold-over plan via WDisc, HDP and GWLP.

4.7 Clarifying the Significance of the Selected Contributions

This section provides important discussion to help readers better appreciate the value of the reviewed work. After considering the foregoing results, the following intuitive questions naturally arise:

- The illustrative example in Table 14 clearly shows the usefulness of the classical fold-over technique (reversing signs + and −) for breaking aliasing and estimating more effects in two-level regular designs. However, one may ask for numerical examples that clarify the utility of the three techniques T_A, T_R, and T_P in improving designs in general cases—regular, non-regular, and with any number of levels. This section addresses the fundamental question: Do these techniques substantially reduce the confounding present in the first-stage experiment, given that they were originally devised for de-aliasing? To answer this, Table 23(source [85]) presents optimal combined designs evaluated via the lower-order effect confounding pattern (LECP)—a widely used measure of factor confounding—and its scalar image obtained with the new DCELF(LECP) criterion from [85], for three- and four-level orthogonal designs. Table 23 (source [85]) shows that the aliasing structure collapses after folding over the original fraction, leaving much less confounding in the combined design. In some orthogonal designs, complete de-aliasing can be achieved through fold-over methods, underscoring the importance of these techniques for reducing or even eliminating confounding-induced interference. It is also noteworthy that the degree of de-aliasing in the combined design depends not only on the initial design; fold-over methods can overcome deficiencies in the first stage and produce an equally good or even better combined design afterward. This phenomenon occurs in several pairs of orthogonal designs, such as $L_1(9,3,4)$ and $L_2(9,3,4)$, $L_1(18,3,7)$ and $L_2(18,3,7)$, etc. Hence, refinements of fold-over techniques are both prominent and promising.
- Given the above discussions, one might ask whether the main contribution of the reviewed work is merely an extension from low- to high-dimensional settings, and whether it constitutes a systematic body of research. The answer is clearly "no" for the following reasons. As the number of levels increases, the problem becomes more difficult, a fact evident from the theoretical results, formulations, and proofs presented in the cited papers—for instance, the formulas and lower bounds for uniformity criteria grow considerably more complex in higher dimensions. Beyond the new efficient expressions and lower bounds that serve as benchmarks for the T_R technique when searching for optimal fold-over plans, each paper introduces novel concepts and perspectives, not simply extensions of earlier ideas. For example: In [92], the T_R technique is introduced for symmetric q-level designs, with corresponding results presented for two-level designs using all discrepancy measures. In [108], the concept of complementary fold-over plans is defined, and results are provided for three-level designs using all discrepancy measures. In [110], the T_R technique is extended to asymmetric mixed q_1- and q_2-level designs, with results given for mixed two- and three-level designs. The results also show that the optimal fold-over plan under LDisc is also optimal under WDisc, and vice versa. In [111], a catalog of optimal fold-over plans for constructing uniform minimum-aberration symmetric four-level combined designs with $2 \le s \le 10$ factors and $8 \le n \le 52$ runs is tabulated, suitable for either qualitative or quantitative factors. In [69], the most general case is treated for the first time. In [85], a new efficient fold-over technique T_A is proposed, and a comprehensive comparison of the three techniques (T_A, T_R, T_P) is presented for the first time. The results show that while the techniques differ significantly in computational complexity, the optimal combined designs they produce are not markedly different. The paper further recommends T_A for its conceptual simplicity and computational advantage. Moreover, it introduces a new criterion (DCELF) that simplifies existing criteria from a sequence to a scalar.
- [69] offers further insightful discussion. After reviewing the results, the following intuitive questions and suggestions for constructing optimal two-stage sequential experimental designs arise:

  Possible strategy 1 [69]: Why fold over an n-run uniform initial design to create a two-stage sequential design with $2n$ runs, when an experimenter could directly select a suitable uniform balanced-levels design table with $2n$ runs from a uniform-design repository?

  Strategy discussion [69]: Unlike one-stage experiments, which require the experimenter to fix all factor settings in advance, two-stage sequential experiments allow the design to be updated and improved after analyzing the data from the initial stage. Consider the following two scenarios:

  Scenario 1 [69]: An experimenter starts with a uniform balanced-levels design of 6 factors at 5 levels and 50 runs, choosing a design $\mathbf{d} \in \mathbb{U}_{50}(5^6)$ as shown in Table 24 (source [69]). After observing the data, the experimenter realizes that the initial design is inadequate and must be supplemented with 50 additional runs. The uniform-design repository (`http://sites.stat.psu.edu/~rli/DMCE/UniformDesign/`) does not contain a uniform design table for $\mathbf{d} \in \mathbb{U}_{100}(5^6)$, and the first-stage experiment with 50 runs has already been conducted. How, then, should the extra 50 runs be chosen? A new approach for constructing uniform or nearly uniform designs that follow an existing initial uniform design is needed. Using the results presented in the cited papers, the optimal fold-over plan is shown to be $\Gamma^* = (2\,2\,0\,1\,2\,2\,0\,2\,0\,0\,0)$, which can generate the 50-run follow-up design $\mathbf{F}^*$ given in Table 24 (source [69]).

The same conclusion holds for the asymmetric design $\mathbf{d} \in \mathbb{U}_{36}(3^{13}4^{12})$ in Table 25 (source [69]). Here the optimal fold-over plan is $\Gamma^* = (1\,0\,2\,2\,0\,2\,1\,0\,2\,0\,0\,0\,1\,1\,1\,3\,3\,0\,0\,3\,1\,2\,0\,3\,1)$, which yields the optimal follow-up design $\mathbf{F}^*$ in Table 25. Its complementary plan $\Gamma^{*c} = (2\,0\,1\,1\,0\,1\,2\,0\,1\,0\,0\,0\,2\,3\,3\,1\,1\,0\,0\,1\,3\,2\,0\,1\,3)$ is also optimal.

Scenario 2 [69]: An experimenter begins with a uniform balanced-levels design of 19 factors at 4 levels and 16 runs, selecting $\mathbf{d}_1 \in \mathbb{U}_{16}(4^{19})$ from the uniform-design repository (Table 26). After analyzing the data, the experimenter decides that the initial design is inadequate and must be supplemented with 16 more runs. The repository does contain a uniform design $\mathbf{d}_2 \in \mathbb{U}_{32}(4^{19})$ for 32 runs (Table 26, source [69]). However, the 32 runs of $\mathbf{d}_2$ are completely different from the original 16 runs of $\mathbf{d}_1$; there is no overlap in level combinations. Consequently, there is no way to extend $\mathbf{d}_1$ by adding 16 runs to obtain the existing 32-run design $\mathbf{d}_2$. This illustrates the need for the methods presented in this paper, which construct uniform or nearly uniform designs by augmenting an existing uniform design with another fraction.

Possible strategy 2 [69]: Why derive new lower bounds for discrepancies based on sequential experimental designs when an experimenter could simply use the lower bounds for balanced-level designs with the same number of runs to evaluate the efficiency of sequential designs?

Strategy discussion [69]: Two-stage sequential experimental designs provide greater precision than one-stage designs with the same total number of runs. If the fold-over structure is ignored, the combined design can be viewed as a balanced-level design, and existing lower bounds for discrepancies apply. However, consider the following scenario:

Scenario 3 [69]: An experimenter starts with a uniform design of 10 factors at 4 levels and 52 runs, selecting $\mathbf{d} \in \mathbb{U}_{52}(4^{10})$ (Table 27, source [69]). Numerical results (Figure 14, source [69]) demonstrate that the lower bound for WDisc based on the sequential (fold-over) design is more useful and sharper than the lower bound based on a balanced design. This shows that the structured approach is valuable and important for obtaining sharper lower bounds for uniformity criteria in sequential experimental designs, which can then serve as benchmarks for finding optimal sequential designs.

Table 23: Orthogonal designs with $3 \le q \le 4$, $3 \le s \le 7$ and $n = 9, 16, 18, 32, 36$. Source [85]

Design	DCELF(LECP)				Design	DCELF(LECP)			
$L(n,q,s)$	Initial	T_A	T_R	T_P	$L(n,q,s)$	Initial	T_A	T_R	T_P
$L(9,3,3)$	7.575428	6.379159*	6.672580	6.379159*	$L_3(32,4,3)$	17.536580	0.000000*	17.493260	0.000000*
$L(18,3,3)$	7.575428	6.379159*	6.672580	6.379159*	$L_4(32,4,3)$	17.493260	0.000000*	17.492619	0.000000*
$L(36,3,3)$	7.575428	6.379159*	6.672580	6.379159*	$L_5(32,4,3)$	17.567611	0.000000*	17.493260	0.000000*
$L_1(9,3,4)$	7.229058	6.672856*	6.773473	6.672856*	$L_6(32,4,3)$	17.536580	0.000000*	17.493260	0.000000*
$L_2(9,3,4)$	7.226499	6.672856*	6.753300	6.672856*	$L_7(32,4,3)$	17.493260	0.000000*	0.000000*	0.000000*
$L(18,3,4)$	7.785251	6.814131*	6.814131*	6.814131*	$L_8(32,4,3)$	17.493260	0.000000*	17.493260	0.000000*
$L_1(18,3,7)$	7.241035	6.883535*	6.979060	6.883535*	$L_9(32,4,3)$	17.493260	0.000000*	0.000000*	0.000000*
$L_2(18,3,7)$	6.937600	6.929135*	6.937600	6.929135*	$L_{10}(32,4,3)$	17.568159	15.378750*	17.566470	15.378750*
$L_1(16,4,3)$	17.493260	9.482329*	9.482329*	9.482329*	$L(16,4,4)$	17.643979	16.711565	17.493261	16.711563*
$L_2(16,4,3)$	17.580901	16.384145*	17.494340	16.384145*	$L_1(16,4,5)$	17.713118	16.733496	17.568833	16.733495*
$L_1(32,4,3)$	17.493260	0.000000*	17.493260	0.000000*	$L_2(16,4,5)$	17.713118	16.733495*	17.421697	16.733495*
$L_2(32,4,3)$	17.567611	0.000000*	17.493722	0.000000*					

[a] The orthogonal designs are from http://www.pietereendebak.nl/oapackage/ and http://neilsloane.com/oadir/.
[b] * refers to the best combined design obtained from different fold-over methods.

Table 24: Initial design and its optimal foirolver design for $\mathbf{d} \in \mathbb{U}_{50}(5^6)$ in Scenario 1. Source [69]

Transpose of the initial design **d**
0 3 1 2 0 0 1 0 0 3 2 3 0 2 0 1 3 2 2 2 0 3 1 1 3 2 3 0 3 1 1 2 3 0 2 3 2 2 3 1 3 1 2 0 1 0 2 1 0 3 1 1
0 3 3 0 2 2 3 3 1 3 2 0 1 2 0 2 1 2 3 1 0 0 1 0 1 1 1 2 3 2 0 2 1 1 1 2 2 0 3 0 0 2 0 3 0 1 3 3 3 2 1 3
1 1 0 1 0 3 2 2 2 2 3 1 3 2 3 3 3 3 0 0 0 2 2 0 0 1 2 1 3 0 1 1 0 1 3 1 2 2 1 3 3 0 0 1 2 2 2 3 1 0 0 3
2 2 2 0 0 1 0 2 3 1 2 1 1 1 2 0 0 3 1 2 1 1 1 2 3 0 2 3 1 3 1 3 0 3 2 0 3 2 3 0 3 1 3 0 3 0 0 3 1 2 0 2
1 3 1 2 1 2 1 1 0 2 0 0 0 1 2 3 1 3 0 1 2 1 1 0 2 2 2 2 2 2 3 0 3 3 3 0 0 3 1 1 0 3 1 0 3 2 3 2 3 3 0 0
0 1 0 1 1 1 1 3 1 3 2 2 0 0 2 2 0 3 1 2 2 3 3 3 3 3 0 1 1 2 0 0 1 3 1 2 3 2 2 3 1 3 1 3 0 2 0 0 2 0 0 2
0 1 2 0 1 1 0 3 2 1 3 2 0 3 2 2 1 2 2 0 1 0 3 1 3 0 2 3 3 0 3 1 2 1 0 3 0 3 1 1 2 2 3 2 1 3 1 2 0 0 3 0
0 2 2 1 1 0 3 1 0 3 2 1 2 3 3 2 3 3 0 3 2 1 3 3 1 2 0 3 1 0 1 3 3 1 0 0 1 0 0 0 2 0 2 2 2 2 0 1 3 2 1 1
1 3 0 0 3 0 1 3 2 2 3 0 2 2 3 3 0 1 3 0 1 3 1 2 1 3 3 0 0 3 3 3 2 2 2 2 0 0 1 2 1 0 2 0 0 1 1 2 2 1 1 1
1 2 3 3 0 3 0 2 1 3 0 0 2 0 1 3 1 2 1 2 3 2 3 0 1 0 3 1 1 2 1 3 2 3 0 3 1 2 0 1 2 0 3 2 0 0 2 0 1 1 2 3
Optimal foldover plan $\Gamma^* = (1\ 0\ 2\ 2\ 0\ 2\ 1\ 0\ 2\ 0\ 0\ 0\ 1\ 1\ 1\ 3\ 3\ 0\ 0\ 3\ 1\ 2\ 0\ 3\ 1)$
Transpose of the optimal foldover design $\mathbf{F}^*$
2 1 3 0 2 2 3 2 2 1 0 1 2 0 2 3 1 0 0 0 2 1 3 3 1 0 1 2 1 3 3 0 1 2 0 1 0 0 1 3 1 3 0 2 3 2 0 3 2 1 3 3
2 1 1 2 0 0 1 1 3 1 0 2 3 0 2 0 3 0 1 3 2 2 3 2 3 3 3 0 1 0 2 0 3 3 3 0 0 2 1 2 2 0 2 1 2 2 1 1 1 0 3 1
1 1 0 1 0 3 2 2 2 2 3 1 3 2 3 3 3 3 0 0 0 2 2 0 0 1 2 1 3 0 1 1 0 1 3 1 2 2 1 3 3 0 0 1 2 2 2 3 1 0 0 3
3 3 3 1 1 2 1 3 0 2 3 2 2 2 3 1 1 0 2 3 2 2 2 3 0 1 3 0 2 0 2 0 1 0 3 1 0 3 0 1 0 2 0 1 0 1 1 0 2 3 1 3
3 1 3 0 3 0 3 3 2 0 2 2 2 3 0 1 3 1 2 3 0 3 3 2 0 0 0 0 0 0 1 2 1 1 1 2 2 1 3 3 2 1 3 2 1 0 1 0 1 1 2 2
2 3 2 3 3 3 3 1 3 1 0 0 2 2 0 0 2 1 3 0 0 1 1 1 1 1 2 3 3 0 2 2 3 1 3 0 1 0 0 1 3 1 3 1 2 0 2 2 0 2 2 0
0 1 2 0 1 1 0 3 2 1 3 2 0 3 2 2 1 2 2 0 1 0 3 1 3 0 2 3 3 0 3 1 2 1 0 3 0 3 1 1 2 2 3 2 1 3 1 2 0 0 3 0
2 0 0 3 3 2 1 3 2 1 0 3 0 1 1 0 1 1 2 1 0 3 1 1 3 1 2 1 3 2 3 1 1 3 2 2 3 2 2 2 0 2 0 0 0 0 2 3 1 0 3 3
1 3 0 0 3 0 1 3 2 2 3 0 2 2 3 3 0 1 3 0 1 3 1 2 1 3 3 0 0 3 3 3 2 2 2 2 0 0 1 2 1 0 2 0 0 1 1 2 2 1 1 1
1 2 3 3 0 3 0 2 1 3 0 0 2 0 1 3 1 2 1 2 3 2 3 0 1 0 3 1 1 2 1 3 2 3 0 3 1 2 0 1 2 0 3 2 0 0 2 0 1 1 2 3

Table 25: Initial design and its optimal foirolver design for $\mathbf{d} \in \mathbb{U}_{36}(3^{13}4^{12})$ in Scenario 1. Source [69]

Transpose of the initial **d**	Optimal foldover plan	Transpose of the optimal foldover design $\mathbf{F}^*$
0 1 2 1 2 0 2 2 2 1 2 1 0 1 0 0 2 2 0 2 1 0 1 2 0 1 2 0 0 1 1 2 0 1 1 0	2	1 2 0 2 0 1 0 0 0 2 0 2 1 2 1 1 0 0 1 0 2 1 2 0 1 2 0 1 1 2 2 0 1 2 2 1
2 2 0 1 1 2 1 0 1 0 2 1 0 2 1 1 0 2 1 2 0 0 1 2 0 1 2 1 2 1 2 0 0 0 2 0	0	2 2 0 1 1 2 1 0 1 0 2 1 0 2 1 1 0 2 1 2 0 0 1 2 0 1 2 1 2 1 2 0 0 0 2 0
1 0 0 0 0 1 1 1 1 0 2 1 1 0 2 0 1 2 1 0 1 0 2 2 0 2 0 2 2 1 2 2 0 2 1 2	1	0 2 2 2 2 0 0 0 0 2 1 0 0 2 1 2 0 1 0 2 0 2 1 1 2 1 2 1 1 0 1 1 2 1 0 1
1 2 2 0 0 0 1 1 2 2 0 1 0 1 2 1 2 2 0 1 0 1 1 1 2 1 0 0 2 2 0 0 2 0 2 1	1	0 1 1 2 2 2 0 0 1 1 2 0 2 0 1 0 1 1 2 0 2 0 0 0 1 2 2 2 1 1 2 2 1 2 1 0
0 0 0 0 2 0 0 2 1 1 0 1 2 2 0 1 0 2 2 2 2 1 0 1 2 2 1 1 2 2 1 1 1 0 1 0	0	0 0 0 0 2 0 0 2 1 1 0 1 2 2 0 1 0 2 2 2 2 1 0 1 2 2 1 1 2 2 1 1 1 0 1 0
2 0 1 1 1 2 2 0 1 2 0 0 0 2 0 0 0 2 1 0 1 1 2 1 2 0 2 2 1 2 0 2 0 1 1 1	1	1 2 0 0 0 1 1 2 0 1 2 2 2 1 2 2 2 1 0 2 0 0 1 0 1 2 1 1 0 1 2 1 2 0 1 0
1 1 1 0 1 1 2 0 2 2 2 1 2 2 2 2 0 1 0 0 1 2 0 1 0 1 0 0 2 2 0 1 1 2 0 0	2	2 2 2 1 2 2 0 1 0 0 0 2 0 0 0 0 1 2 1 1 2 0 1 2 1 2 2 1 0 0 1 2 2 0 1 1
2 2 1 1 0 1 0 1 2 1 0 1 0 1 1 0 2 0 1 2 0 2 0 1 1 2 0 2 2 2 1 2 0 2 0 0	0	2 2 1 1 0 1 0 1 2 1 0 1 0 1 1 0 2 0 1 2 0 2 0 1 1 2 0 2 2 2 1 2 0 2 0 0
1 0 2 0 1 2 0 0 1 2 2 0 1 1 0 2 1 0 2 2 0 0 1 1 1 2 0 0 0 2 1 2 1 1 2 2	1	0 2 1 2 0 1 2 2 0 1 1 2 0 0 2 1 0 2 1 1 2 1 0 1 1 1 2 2 2 1 0 1 0 0 1 1
1 1 2 1 0 0 0 1 1 1 1 0 2 2 2 1 2 1 2 0 1 2 2 1 0 2 2 1 0 0 0 2 0 0 2 0	0	1 1 2 1 0 0 0 1 1 1 1 0 2 2 2 1 2 1 2 0 1 2 2 1 0 2 2 1 0 0 0 2 0 0 2 0
0 1 1 2 1 2 0 1 2 0 1 2 2 2 0 2 2 2 0 0 0 1 1 0 1 0 0 1 2 1 1 2 0 0 1 2	0	0 1 1 2 1 2 0 1 2 0 1 2 2 2 0 2 2 2 0 0 0 1 1 0 1 0 0 1 2 1 1 2 0 0 1 2
2 1 2 0 2 0 1 0 0 2 0 2 2 0 2 1 1 2 1 2 1 0 1 1 1 0 0 2 1 0 1 1 0 2 2 0	0	2 1 2 0 2 0 1 0 0 2 0 2 2 0 2 1 1 2 1 2 1 0 1 1 1 0 0 2 1 0 1 1 0 2 2 0
1 1 2 0 1 0 0 1 2 1 2 2 2 1 2 1 0 0 0 2 2 2 0 2 0 0 0 1 2 2 1 1 1 2 0 0 1	2	2 2 0 1 2 1 1 2 0 2 0 0 2 0 2 1 1 1 0 0 0 1 0 1 1 1 0 1 0 2 2 2 0 1 1 2
- -	-	- -
0 3 1 0 1 2 2 1 1 3 3 2 0 2 0 2 0 1 3 2 3 0 3 1 0 1 3 0 2 2 3 1 1 3 0 2	3	1 0 2 1 2 3 3 2 2 0 0 3 1 3 1 3 1 2 0 3 0 1 0 2 1 2 0 1 3 3 0 2 2 0 1 3
3 1 2 3 1 1 0 3 0 3 3 0 0 2 1 3 0 0 0 1 2 1 1 3 3 1 0 2 0 2 2 2 1 2 2 3	3	0 2 3 0 2 2 1 0 1 0 0 1 1 3 2 0 1 1 1 2 3 2 2 0 0 2 1 3 1 3 3 3 2 3 3 0
0 3 1 3 3 2 1 2 1 1 3 2 0 3 3 1 2 3 2 0 2 1 1 0 2 0 0 2 0 1 3 3 2 0 1 0	1	3 2 0 2 2 1 0 1 0 0 2 1 3 2 2 0 1 2 1 3 1 0 0 3 1 3 3 1 3 0 2 2 1 3 0 3
2 1 2 1 3 2 2 0 1 3 2 0 0 0 2 1 3 3 0 0 2 1 0 0 1 3 2 3 1 3 1 1 2 0 3 3	1	1 0 1 0 2 1 1 3 0 2 1 3 3 3 1 0 2 2 3 3 1 0 3 3 0 2 1 2 0 2 0 0 1 3 2 2
2 2 1 1 0 3 1 0 0 1 3 2 2 1 1 3 3 3 1 0 0 0 2 3 2 1 0 0 0 2 2 3 2 3 3 1	0	2 2 1 1 0 3 1 0 0 1 3 2 2 1 1 3 3 3 1 0 0 0 2 3 2 1 0 0 0 2 2 3 2 3 3 1
1 0 3 3 1 0 1 2 0 2 2 3 0 1 3 2 1 3 0 3 1 1 3 1 2 2 3 0 2 0 1 0 2 2 3 0	0	1 0 3 3 1 0 1 2 0 2 2 3 0 1 3 2 1 3 0 3 1 1 3 1 2 2 3 0 2 0 1 0 2 2 3 0
3 0 0 2 0 3 1 2 3 3 0 0 1 3 3 1 3 1 1 0 1 2 2 0 0 1 2 0 3 2 1 2 2 3 1 2	1	2 3 3 1 3 2 0 1 2 2 3 3 0 2 2 0 2 0 0 3 0 1 1 3 3 0 1 3 2 1 0 1 1 2 0 1
0 3 2 0 0 1 2 2 1 2 2 3 2 1 3 3 2 3 0 2 3 3 1 0 1 1 1 1 0 0 1 2 0 3 0 3	3	1 0 3 1 1 2 3 3 2 3 3 0 3 2 0 0 3 0 1 3 0 0 2 1 2 2 2 2 1 1 2 3 1 0 1 0
1 3 3 2 1 2 1 1 0 2 0 2 2 3 1 1 0 3 0 0 2 3 1 3 3 2 3 1 2 3 0 2 0 1 0 0	2	3 1 1 0 3 0 3 3 2 0 2 0 0 1 3 3 2 1 2 2 0 1 3 1 1 0 1 3 0 1 2 0 2 3 2 2
0 1 1 1 3 0 3 0 2 0 2 3 2 2 2 1 1 0 0 0 2 0 2 2 3 1 3 1 1 1 0 3 2 3 3 3	0	0 1 1 1 3 0 3 0 2 0 2 3 2 2 2 1 1 0 0 0 2 0 2 2 3 1 3 1 1 1 0 3 2 3 3 3
0 2 1 3 2 1 3 1 3 2 0 0 2 2 1 2 2 3 0 3 3 0 0 1 3 0 2 0 1 3 1 0 2 3 1 1	1	3 1 0 2 1 0 2 0 2 1 3 3 1 1 0 1 1 2 3 2 2 3 3 0 2 3 0 3 0 2 0 3 1 2 0 0
2 2 3 3 3 3 0 2 2 0 3 2 3 2 1 1 1 0 1 1 2 3 0 0 1 2 1 0 0 1 0 1 0 3 2 3	3	3 3 0 0 0 0 1 3 3 1 0 3 0 3 2 2 2 1 2 2 3 0 1 1 2 3 3 1 1 2 1 2 1 0 3 0

Table 26: The two uniform designs for Scenario 2. Source [69]

Runs No.	$\mathbf{d}_1$	$\mathbf{d}_2$
1	1 0 1 3 1 0 2 2 3 0 0 2 1 0 1 0 1 3	1 3 1 2 2 0 2 3 3 3 2 3 3 1 3 2 0 0
2	1 3 0 3 0 1 1 1 2 2 3 3 3 1 3 3 3 3	1 1 3 3 3 2 1 1 1 3 2 2 0 0 3 1 1 3
3	0 0 2 0 3 2 2 1 0 1 3 2 2 2 0 3 0 0	3 0 2 3 2 3 0 2 3 2 2 2 2 2 0 3 3 3
4	3 0 2 0 1 0 0 3 2 2 2 1 2 0 0 2 3 2	2 2 2 2 3 2 0 0 2 3 3 1 1 3 1 2 0 3
5	1 2 2 1 1 2 3 2 0 1 2 0 3 0 3 0 2 1	0 3 3 1 0 0 0 0 2 1 3 2 0 2 3 3 0 1
6	0 1 1 2 1 1 0 3 1 0 0 3 0 2 3 2 0 1	0 2 0 2 2 1 0 1 3 2 0 0 0 2 0 1 2 0
7	3 1 3 0 0 1 2 0 3 0 1 0 3 2 1 3 2 2	3 3 0 3 3 3 1 3 1 2 1 1 3 3 3 0 0 1
8	2 2 0 3 2 0 0 1 1 1 1 0 1 2 0 1 2 3	1 0 1 3 0 1 3 1 0 0 1 1 1 0 0 3 0 0
9	1 3 2 2 3 1 3 0 2 3 0 1 0 3 0 1 2 0	1 2 1 1 1 3 0 1 3 0 2 1 3 3 2 0 2 2
10	2 1 1 1 0 3 3 1 1 3 2 2 3 3 1 2 0 3	2 1 3 2 1 1 0 3 0 3 1 2 3 1 1 3 2 1
11	3 0 0 1 2 3 1 0 3 2 0 0 1 3 3 0 0 1	0 3 3 3 1 0 2 0 3 3 3 1 2 0 1 0 1 2
12	3 3 0 0 3 2 1 2 1 0 1 2 0 0 2 1 3 1	0 1 1 3 0 3 0 3 1 0 3 0 2 2 2 2 1 0
13	0 2 1 3 2 3 1 0 0 3 2 1 0 1 2 3 1 2	1 1 0 0 3 3 3 1 3 1 0 2 2 1 1 2 0 2
14	0 1 3 2 2 3 3 2 2 2 1 3 2 1 1 1 1 0	2 2 1 1 0 0 2 3 0 2 0 2 1 2 0 0 1 3
15	2 3 3 1 0 2 0 3 3 1 3 1 1 1 2 2 1 0	1 1 0 0 1 2 2 0 1 2 3 3 3 3 1 3 3 0
16	2 2 3 2 3 0 2 3 0 3 3 3 2 3 2 0 3 2	0 0 2 1 0 2 1 3 3 3 1 3 0 3 0 2 2 2
17		2 3 1 0 3 2 1 0 0 1 2 0 2 1 0 1 2 1
18		3 2 1 3 3 2 2 2 2 0 0 3 0 0 1 3 2 1
19		2 0 0 2 0 2 1 2 3 1 3 1 0 1 2 0 3 0
20		2 0 2 2 3 0 2 1 1 1 0 0 3 3 3 3 1 2
21		3 2 3 2 0 1 3 2 0 2 2 3 2 3 2 1 0 2
22		3 3 1 1 1 1 1 1 1 2 3 2 1 1 2 2 3 2
23		0 0 0 1 3 0 3 2 1 3 2 0 1 2 1 0 3 1
24		0 3 2 0 2 1 1 2 0 0 0 1 3 2 1 1 0 3
25		0 1 2 2 2 3 2 0 0 1 1 2 1 0 2 1 3 1
26		1 2 0 1 2 1 1 3 2 1 1 0 2 0 2 3 1 3
27		2 2 3 0 2 3 3 3 0 0 3 0 0 0 3 2 3 2
28		3 1 2 0 0 1 3 0 2 3 0 0 3 1 3 0 3 3
29		3 1 3 0 1 0 2 2 2 1 2 1 1 3 0 2 2 0
30		1 0 2 0 1 3 0 2 2 2 0 3 1 0 3 1 1 1
31		3 3 3 1 2 2 3 1 1 0 1 3 2 1 0 0 1 0
32		2 0 0 3 1 0 3 0 2 0 1 3 0 2 2 1 2 3

Table 27: Intial design $\mathbf{d} \in \mathbb{U}_{52}(4^{10})$ for Scenario 3. Source [69]

Initial design $\mathbf{d}^\top$
0 3 1 2 0 0 1 0 0 3 2 3 0 2 0 1 3 2 2 2 0 3 1 1 3 2 3 0 3 1 1 2 3 0 2 3 2 2 3 1 3 1 2 0 1 0 2 1 0 3 1 1
0 3 3 0 2 2 3 3 1 3 2 0 1 2 0 2 1 2 3 1 0 0 1 0 1 1 1 2 3 2 0 2 1 1 1 2 2 0 3 0 0 2 0 3 0 1 3 3 3 2 1 3
1 1 0 1 0 3 2 2 2 2 3 1 3 2 3 3 3 3 0 0 0 2 2 0 0 1 2 1 3 0 1 1 0 1 3 1 2 2 1 3 3 0 0 1 2 2 2 3 1 0 0 3
2 2 2 0 0 1 0 2 3 1 2 1 1 1 2 0 0 3 1 2 1 1 1 2 3 0 2 3 1 3 1 3 0 3 2 0 3 2 3 0 3 1 3 0 3 0 0 3 1 2 0 2
1 3 1 2 1 2 1 1 0 2 0 0 0 1 2 3 1 3 0 1 2 1 1 0 2 2 2 2 2 2 3 0 3 3 3 0 0 3 1 1 0 3 1 0 3 2 3 2 3 3 0 0
0 1 0 1 1 1 1 3 1 3 2 2 0 0 2 2 0 3 1 2 2 3 3 3 3 3 0 1 1 2 0 0 1 3 1 2 3 2 2 3 1 3 1 3 0 2 0 0 2 0 0 2
0 1 2 0 1 1 0 3 2 1 3 2 0 3 2 2 1 2 2 0 1 0 3 1 3 0 2 3 3 0 3 1 2 1 0 3 0 3 1 1 2 2 3 2 1 3 1 2 0 0 3 0
0 2 2 1 1 0 3 1 0 3 2 1 2 3 3 2 3 3 0 3 2 1 3 3 1 2 0 3 1 0 1 3 3 1 0 0 1 0 0 0 2 0 2 2 2 2 0 1 3 2 1 1
1 3 0 0 3 0 1 3 2 2 3 0 2 2 3 3 0 1 3 0 1 3 1 2 1 3 3 0 0 3 3 3 2 2 2 2 0 0 1 2 1 0 2 0 0 1 1 2 2 1 1 1
1 2 3 3 0 3 0 2 1 3 0 0 2 0 1 3 1 2 1 2 3 2 3 0 1 0 3 1 1 2 1 3 2 3 0 3 1 2 0 1 2 0 3 2 0 0 2 0 1 1 2 3

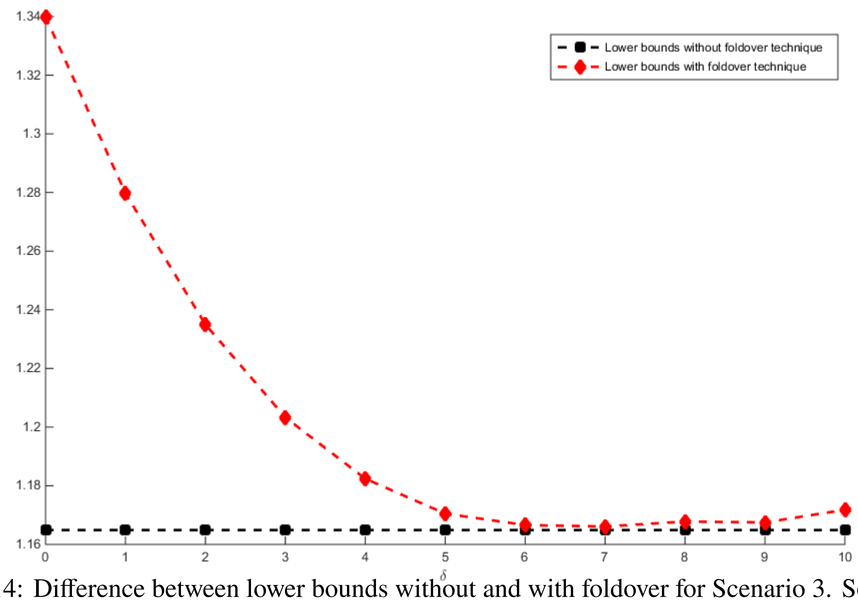


Figure 14: Difference between lower bounds without and with foldover for Scenario 3. Source [69]

5 Low-Complexity Non-Searching Techniques for Constructing Optimal Designs

As noted earlier, several techniques are available for constructing optimal experimental designs, yet the challenge faced by experimenters remains substantial. Experimental designs involving a large number of factors and runs are essential in many practical applications. For example, [114] examined a deterministic simulation of the CO_2 greenhouse effect with 281 factors (see also [115]). In another case, [116] applied sequential bifurcation to a discrete-event simulation of a supply chain at Ericsson in Sweden, involving 92 factors. Likewise, stochastic simulations of computer and telecommunication networks often incorporate hundreds of factors, such as service rates, arrival rates, and numbers of servers. [117] also investigated multiple simulation models with hundreds of factors. Given these demanding applications, there is a clear and significant need to extend the existing tripling technique into a more general multiple tripling framework. Consequently, there is a clear need to develop theoretical construction techniques that yield optimal designs even in special cases. Recently, promising attempts have been made for rather specialized settings. The following subsections summarize the main findings; interested readers may consult the cited papers for further details and theoretical derivations.

5.1 Multiple Doubling Algorithm for Constructing Large Two-Level Designs

[118] introduced the *Multiple Doubling Algorithm (MDA)* as an efficient theoretical construction method for optimal two-level designs. Doubling [119] is a simple classical technique for enlarging a two-level design. For any $n \times s$ matrix X with entries -1 and 1, the double of X is the $2n \times 2s$ matrix

$$D(X) = \begin{pmatrix} X & X \\ X & -X \end{pmatrix}.$$

[119] employed the doubling method to construct orthogonal main-effect plans (orthogonal designs). [120] discussed the construction of two-level FrFDs of resolution IV by doubling regular two-level FrFDs of resolution III. A general complementary-design theory for doubling was investigated in [121]. The uniformity of the double FrFDs is discussed in [122, 123]. *Although the doubling technique has been thoroughly studied, no previous work had systematically addressed multiple (more than one-time) doubling*. Multiple doubling of two-level designs is essential for many real-world experiments that demand a large number of runs and factors. For any two-level design $\mathbf{d} \in \mathbb{U}_n(2^s)$ with entries 0 and 1 and any integer $m \geq 1$, the m-double design generated from $\mathbf{d}$ is defined in [118] as:

$$\mathbf{D}^{\odot^m} = \underbrace{\odot \cdots \odot}_{m \text{ times}} \mathbf{d} \ \in \ \mathbb{U}_{2^m n}(2^{2^m s}), \qquad \odot(\mathbf{d}) = \begin{pmatrix} \mathbf{d} & \mathbf{d} \\ \mathbf{d} & \mathbf{1_d} - \mathbf{d} \end{pmatrix}, \tag{16}$$

where $\mathbf{1_d}$ is an all-ones matrix of the same dimensions as $\mathbf{d}$. This process increases the design size exponentially while preserving its structural integrity.

Example 5.1 *[118] Doubling of the design $\boldsymbol{d} = (0\ 1)^\top \in \mathbb{U}_2(2^2)$ four times is a design $\boldsymbol{D}^{\odot^4} \in \mathbb{U}_{32}(2^{32})$ which is given as follows*

$$\boldsymbol{D}^{\odot^4} = \begin{pmatrix}
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0 \\
0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0 \\
0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1 \\
0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0 \\
0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1 \\
0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1 \\
0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0 \\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0 \\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1 \\
0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1 \\
0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0 \\
0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&0&1&0&1&0&1&0&1 \\
0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&0&1&1&0&0&1&1&0 \\
0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&1&0&1&0&0&1&0&1&0&1&0&1&1&0&1&0 \\
0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0&1&0&0&1&0&1&1&0&0&1&1&0&1&0&0&1
\end{pmatrix}^\top.$$

Remark 1 ([118]) *To construct a two-level design with 256 factors and 512 runs using the classical (one-step) doubling technique, one first needs an initial two-level FrFD with 128 factors and 256 runs. A primary difficulty is then how to select that initial design. By contrast, a uniform two-level orthogonal array of strength 3 with 256 factors and 512 runs can be obtained directly by doubling the simple vector ($\mathbf{d} = (1\ 0)^\top$) eight times, or by doubling a FuFD with two factors seven times (see Table 28, source [118]). Hence, the MDA provides a simple and effective route to optimal two-level FrFDs of large size.*

[118] examined the statistical properties of the resulting multiple-doubling designs from the four optimization perspectives mentioned earlier. Several natural questions arise: (1) How can initial designs be chosen effectively to yield optimal multiple-double designs? (2) How can one measure the optimality of a multiple-double design relative to all designs of the same size? (3) How does the multiple-doubling technique compare in efficiency with other widely used methods for constructing large two-level designs? Through theoretical and computational analysis, [118] answered these questions. Without any computer search, the authors produced new recommended optimal designs that either outperform existing recommendations or cannot be built by existing techniques because of their large dimensions. Complex theoretical relationships between the generated designs and the kernel designs are detailed in [118]. Key conclusions regarding the efficiency of the new technique include:

- For FuFDs with $2 \leq s \leq 7$ factors, Tables A13-A18 in the online supplement of [118] list the LDisc values of all generated $3 \leq m \leq 20$-double designs $\mathbf{D}_s^{\odot^m} \in \mathbb{U}_{2^{s+m}}(2^{2^m s})$ together with the corresponding lower bounds of the LDisc for designs of the same size. If the LDisc value of a generated m-double design equals its lower bound, that design attains the minimum possible LDisc among all designs in $\mathbb{U}_{2^{s+m}}(2^{2^m s})$ and is therefore a *uniform design*. Table 28 (source [118]) summarizes the results from those supplementary tables. From it, [118] concluded that all generated $3 \leq m \leq 20$-double designs are uniform orthogonal arrays of strength 3 within the set $\mathbb{U}_{2^{s+m}}(2^{2^m s})$.
- The two-level orthogonal arrays of strength 3, $\mathbf{d} \in \mathbb{U}_{2^{a+1}}(2^{2^a})$ for $2 \leq a \leq 8$, available on the website `http://neilsloane.com/oadir/`, can be generated from the initial two-level FuFD $\mathbf{d} \in \mathbb{U}_4(2^2)$ via the MDA without any computer search, yielding $1 \leq m \leq 7$-double designs $\mathbf{D}^{\odot^m} \in \mathbb{U}_{2^{m+2}}(2^{2^{m+1}})$. These designs can also be obtained by $2 \leq m \leq 8$ doublings of $\mathbf{d} = (0\ 1)^\top$. The 20-double design $\mathbf{D}^{\odot^{20}} \in \mathbb{U}_{4194304}(2^{2097152})$ in Table 28, generated from the initial FuFD $\mathbf{d} \in \mathbb{U}_2(2^2)$ with the MDA in zero time, is a uniform design under the LDisc criterion. This design corresponds to 21 doublings of $\mathbf{d} = (0\ 1)^\top$. By contrast, the (adjusted) threshold-accepting algorithm would require several days merely to compare the LDisc values of the millions of possible designs in $\mathbb{U}_{4194304}(2^{2097152})$ to locate a global or local minimum. The level-permutation algorithm would likewise take days to select the best design from all level-permuted versions of any initial design in that space. The factor-projection algorithm would need several days to choose the best sub-design with 2,097,152 factors from an initial design with even more factors. Augmented-design techniques would require repeatedly folding over a uniform design and comparing millions of candidate augmented designs. Moreover, none of those alternative methods guarantees that the resulting design will be uniform.

Table 28: The conclusions of the results in the Tables A13-A18 in the supplementary material for m-double designs $\mathbf{D}_s^{\odot^m} \in \mathbb{U}_{2^{s+m}}(2^{2^m s})$, $3 \leq t \leq 20$ generated from FuFDs with $2 \leq s \leq 5$ factors. Source [118]

♯ doubling	s	♯ factors	♯ runs	Optimality	s	♯ factors	♯ runs	Optimality
3	2	16	32	Uniform orthogonal array of strength 3	3	24	64	Uniform orthogonal array of strength 3
4	2	32	64	Uniform orthogonal array of strength 3	3	48	128	Uniform orthogonal array of strength 3
5	2	64	128	Uniform orthogonal array of strength 3	3	396	256	Uniform orthogonal array of strength 3
6	2	128	256	Uniform orthogonal array of strength 3	3	192	512	Uniform orthogonal array of strength 3
7	2	256	512	Uniform orthogonal array of strength 3	3	384	1024	Uniform orthogonal array of strength 3
8	2	512	1024	Uniform orthogonal array of strength 3	3	768	2048	Uniform orthogonal array of strength 3
9	2	1024	2048	Uniform orthogonal array of strength 3	3	1536	8192	Uniform orthogonal array of strength 3
10	2	2048	4096	Uniform orthogonal array of strength 3	3	3072	4096	Uniform orthogonal array of strength 3
11	2	4096	8192	Uniform orthogonal array of strength 3	3	6144	16384	Uniform orthogonal array of strength 3
12	2	8192	16384	Uniform orthogonal array of strength 3	3	12288	32768	Uniform orthogonal array of strength 3
13	2	16384	32768	Uniform orthogonal array of strength 3	3	24576	65536	Uniform orthogonal array of strength 3
14	2	32768	65536	Uniform orthogonal array of strength 3	3	49152	131072	Uniform orthogonal array of strength 3
15	2	65536	131072	Uniform orthogonal array of strength 3	3	98304	262144	Uniform orthogonal array of strength 3
16	2	131072	262144	Uniform orthogonal array of strength 3	3	196608	524288	Uniform orthogonal array of strength 3
17	2	262144	524288	Uniform orthogonal array of strength 3	3	393216	1048576	Uniform orthogonal array of strength 3
18	2	524288	1048576	Uniform orthogonal array of strength 3	3	786432	2097152	Uniform orthogonal array of strength 3
19	2	1048576	2097152	Uniform orthogonal array of strength 3	3	1572864	4194304	Uniform orthogonal array of strength 3
20	2	2097152	4194304	Uniform orthogonal array of strength 3	3	3145728	8388608	Uniform orthogonal array of strength 3
3	4	32	128	Uniform orthogonal array of strength 3	5	40	256	Uniform orthogonal array of strength 3
4	4	64	256	Uniform orthogonal array of strength 3	5	80	512	Uniform orthogonal array of strength 3
5	4	128	512	Uniform orthogonal array of strength 3	5	160	1024	Uniform orthogonal array of strength 3
6	4	256	1024	Uniform orthogonal array of strength 3	5	320	2048	Uniform orthogonal array of strength 3
7	4	512	2048	Uniform orthogonal array of strength 3	5	640	4096	Uniform orthogonal array of strength 3
8	4	1024	40964	Uniform orthogonal array of strength 3	5	1280	8192	Uniform orthogonal array of strength 3
9	4	2048	8192	Uniform orthogonal array of strength 3	5	2560	16384	Uniform orthogonal array of strength 3
10	4	4096	16384	Uniform orthogonal array of strength 3	5	5120	32768	Uniform orthogonal array of strength 3
11	4	8192	32768	Uniform orthogonal array of strength 3	5	10240	65536	Uniform orthogonal array of strength 3
12	4	16384	65536	Uniform orthogonal array of strength 3	5	20480	131072	Uniform orthogonal array of strength 3
13	4	32768	131072	Uniform orthogonal array of strength 3	5	40960	262144	Uniform orthogonal array of strength 3
14	4	65536	262144	Uniform orthogonal array of strength 3	5	81920	524288	Uniform orthogonal array of strength 3
15	4	131072	524288	Uniform orthogonal array of strength 3	5	163840	1048576	Uniform orthogonal array of strength 3
16	4	262144	1048576	Uniform orthogonal array of strength 3	5	327680	2097152	Uniform orthogonal array of strength 3
17	4	524288	2097152	Uniform orthogonal array of strength 3	5	655360	4194304	Uniform orthogonal array of strength 3
18	4	1048576	4194304	Uniform orthogonal array of strength 3	5	1310720	8388608	Uniform orthogonal array of strength 3
19	4	2097152	8388608	Uniform orthogonal array of strength 3	5	2621440	16777216	Uniform orthogonal array of strength 3
20	4	4194304	16777216	Uniform orthogonal array of strength 3	5	5242880	33554432	Uniform orthogonal array of strength 3

5.2 Multiple Tripling Algorithm for Constructing Large Three-Level Designs

[124] extended the idea of the Multiple Doubling Algorithm (MDA) for two-level designs $\mathbf{d} \in \mathbb{U}_n(2^s)$ to a *Multiple Tripling Algorithm (MTA)* for three-level designs $\mathbf{d} \in \mathbb{U}_n(3^s)$. For any design $\mathbf{d} \in \mathbb{U}_n(3^s)$, let $\mathbf{T}^{\uplus^0} = \mathbf{d}$ and $1 \leq \eta \leq m$. The operation of tripling $\mathbf{d}$ m times produces an m-triple design, denoted $\mathbf{T}^{\uplus^m} \in \mathbb{U}_{3^m n}(3^{3^m s})$ and defined in [124] as:

$$\mathbf{T}^{\uplus^m} = \underbrace{\uplus \cdots \uplus}_{m \text{ times}} \mathbf{d} \in \mathbb{U}_{3^m n}\left(3^{3^m s}\right), \qquad \uplus(\mathbf{d}) = \begin{pmatrix} \mathbf{d} & \mathbf{d} & \circlearrowleft_1 \mathbf{d} \\ \mathbf{d} & \circlearrowleft_4 \mathbf{d} & \circlearrowleft_2 \mathbf{d} \\ \mathbf{d} & \circlearrowleft_5 \mathbf{d} & \circlearrowleft_3 \mathbf{d} \end{pmatrix},$$

where $\circlearrowleft_i$ $(0 \leq i \leq 5)$ are all possible level permutations of the three-level codes $(0, 1, 2)$, as given in Table 29. When $m = 1$, [125] studied the one-time tripling technique.

Table 29: All the possible six level permutations of (0 1 2). Source [124]

Code	Level permutation	Image	Code	Level permutation	Image
(0 1 2)	$\xrightarrow{\circlearrowleft_0}$	(0 1 2)	(0 1 2)	$\xrightarrow{\circlearrowleft_3}$	(1 0 2)
(0 1 2)	$\xrightarrow{\circlearrowleft_1}$	(0 2 1)	(0 1 2)	$\xrightarrow{\circlearrowleft_4}$	(2 0 1)
(0 1 2)	$\xrightarrow{\circlearrowleft_2}$	(2 1 0)	(0 1 2)	$\xrightarrow{\circlearrowleft_5}$	(1 2 0)

Example 5.2 *[124]Tripling of a very simple three-level design with one factor $\boldsymbol{d} = (0\ 1\ 2)^\top$ three times by the technique MTA gives the following orthogonal three-level design $\boldsymbol{T}^{\uplus^3} \in \mathbb{U}_{81}(3^{27})$.*

$$\boldsymbol{T}^{\uplus^3} = \begin{pmatrix}
012012012012012012012012012012012012012012012012012012012012012012012012012012012 \\
012201120012201120012201120012201120012201120012201120012201120012201120012201120 \\
021210102021210102021210102021210102021210102021210102021210102021210102021210102 \\
012012012201201201120120120012012012201201201120120120012012012201201201120120120 \\
012201120201120012120012201012201120201120012120012201012201120201120012120012201 \\
021210102210102021102021210021210102210102021102021210021210102210102021102021210 \\
021021021210210210102102102021021021210210210102102102021021021210210210102102102 \\
021102210210021102102210021021102210210021102102210021021102210210021102102210021 \\
012120201201012120120201012012120201201012120120201012012120201201012120120201012 \\
012012012012012012012012012201201201201201201201201201120120120120120120120120120 \\
012201120012201120012201120201120012201120012201120012120012201120012201120012201 \\
021210102021210102021210102210102021210102021210102021102021210102021210102021210 \\
012012012201201201120120120201201201120120120012012012120120120012012012201201201 \\
012201120201120012120012201201120012120012201012201120120012201012201120201120012 \\
021210102210102021102021210210102021102021210021210102102021210021210102210102021 \\
021021021210210210102102102210210210102102102021021021102102102021021021210210210 \\
021102210210021102102210021210021102102210021021102210102210021021102210210021102 \\
012120201201012120120201012201012120120201012012120201120201012012120201201012120 \\
021021021021021021021021021210210210210210210210210210102102102102102102102102102 \\
021102210021102210021102210210021102210021102210021102102210021102210021102210021 \\
012120201012120201012120201201012120201012120201012120120201012120201012120201012 \\
021021021102102102210210210210210210021021021102102102102102102210210210021021021 \\
021102210102210021210021102210021102021102210102210021102210021210021102021102210 \\
012120201120201012201012120201012120012120201120201012120201012201012120012120201 \\
012012012120120120201201201201201201012012012120120120120120120201201201012012012 \\
012201120120012201201120012201120012012201120120012201120012201201120012012201120 \\
021210102102021210210102021210102021021210102102021210102021210210102021021210102
\end{pmatrix}^\top.$$

[124] studied the statistical properties of the resulting multiple-triple designs from the four optimization perspectives mentioned earlier. Complex theoretical relationships between the generated designs and the kernel designs are detailed in [124]. Key conclusions regarding the efficiency of the new technique include:

- For full factorial designs (FuFDs) with $2 \leq s \leq 3$ factors, Tables 30 and 31 (source [124]) list, for each resulting $3 \leq m \leq 15$-triple design $\mathbf{T}_s^{\uplus^m} \in \mathbb{U}_{3^{s+m}}(3^{3^m s})$, the number of tripling steps, the number of runs, the number of factors, the LDisc value, the corresponding lower bound from [124], and the aberration measures $\mathcal{A}_1$ and $\mathcal{A}_2$. These tables show that all generated $3 \leq m \leq 15$-triple designs are orthogonal uniform designs.
- The new multiple-tripling technique constructs the above-mentioned recommended optimal designs in zero time (i.e., without computer search). By contrast, established techniques such as the threshold-accepting algorithm, level-permutation algorithms, and factor-projection algorithms require considerable time to produce designs of comparable size, as noted earlier.

Furthermore, [126] enhanced the generated multiple-triple designs by applying a level-permutation algorithm. That work presents many intricate theoretical relationships between the initial designs and their level-permuted versions, which are used to improve the quality of the resulting three-level designs.

Table 30: Multiple tripling of the three-level FuFDs with two factors. Source [124]

m	♯ factors	♯ runs	LDisc-value	Lower bound	$\mathcal{A}_1(\mathbf{d}^{\uplus^m})$	$\mathcal{A}_2(\mathbf{T}^{\uplus^m})$	Optimality
3	54	243	0.00411566	0.00411564	0	0	Uniform orthogonal design
4	162	729	0.0013717421	0.0013717421	0	0	Uniform orthogonal design
5	486	2187	0.0004572474	0.0004572474	0	0	Uniform orthogonal design
6	1458	6561	0.0001524158	0.0001524158	0	0	Uniform orthogonal design
7	4374	19683	0.0000508053	0.0000508053	0	0	Uniform orthogonal design
8	13122	59049	0.0000169351	0.0000169351	0	0	Uniform orthogonal design
9	39366	177147	0.000005645	0.000005645	0	0	Uniform orthogonal design
10	118098	531441	0.0000018817	0.0000018817	0	0	Uniform orthogonal design
11	354294	1594323	0.0000006272	0.0000006272	0	0	Uniform orthogonal design
12	1062882	4782969	0.0000002091	0.0000002091	0	0	Uniform orthogonal design
13	3188646	14348907	0.0000000697	0.0000000697	0	0	Uniform orthogonal design
14	9565938	43046721	0.0000000232	0.0000000232	0	0	Uniform orthogonal design
15	28697814	129140163	0.0000000077	0.0000000077	0	0	Uniform orthogonal design

Table 31: Multiple tripling of the three-level FuFDs with three factors. Source [124]

m	♯ factors	♯ runs	LDisc-value	Lower bound	$\mathcal{A}_1(\mathbf{T}^{\uplus^m})$	$\mathcal{A}_2(\mathbf{T}^{\uplus^m})$	Optimality
3	81	729	0.0013718	0.0013717424	0	0	Uniform orthogonal design
4	243	2187	0.0004572474	0.0004572474	0	0	Uniform orthogonal design
5	729	6561	0.0001524158	0.0001524158	0	0	Uniform orthogonal design
6	2187	19683	0.0000508053	0.0000508053	0	0	Uniform orthogonal design
7	6561	59049	0.0000169351	0.0000169351	0	0	Uniform orthogonal design
8	19683	177147	0.000005645	0.000005645	0	0	Uniform orthogonal design
9	59049	531441	0.0000018817	0.0000018817	0	0	Uniform orthogonal design
10	177147	1594323	0.0000006272	0.0000006272	0	0	Uniform orthogonal design
11	531441	4782969	0.0000002091	0.0000002091	0	0	Uniform orthogonal design
12	1594323	14348907	0.0000000697	0.0000000697	0	0	Uniform orthogonal design
13	4782969	129140163	0.0000000232	0.0000000232	0	0	Uniform orthogonal design
14	14348907	43046721	0.0000000077	0.0000000077	0	0	Uniform orthogonal design
15	43046721	387420489	0.0000000026	0.0000000026	0	0	Uniform orthogonal design

5.3 Multiple Quadrupling Algorithm for Constructing Large Four-Level Designs

[127] extended the ideas of the MDA for two level designs $\mathbf{d} \in \mathbb{U}_n(2^s)$ and MTA $\mathbf{d} \in \mathbb{U}_n(3^s)$ to *Multiple Quadrupling Algorithm (MQA)* for four-level designs $\mathbf{d} \in \mathbb{U}_n(4^s)$. For any design $\mathbf{d} \in \mathbb{U}_n(4^s)$, let $\mathbf{Q}^{\otimes^0} = \mathbf{d}$ and $1 \leq \eta \leq m$. Quadrupling of $\mathbf{d}$ m times is called m-quadruple design, denoted by $\mathbf{Q}^{\otimes^m} \in \mathbb{U}_{4^m n}(4^{4^m s})$ and defined in [127] as follows

$$\mathbf{Q}^{\otimes^m} = \underbrace{\otimes \cdots \otimes}_{m\text{times}} \mathbf{d} \in \mathbb{U}_{4^m n}(4^{4^m s}), \qquad \otimes(\mathbf{d}) = \begin{pmatrix} \mathbf{d} & \mathbf{d} & \mathbf{d} & \mathbf{d} \\ \mathbf{d} & \circlearrowleft_1 \mathbf{d} & \circlearrowleft_2 \mathbf{d} & \circlearrowleft_3 \mathbf{d} \\ \mathbf{d} & \circlearrowleft_2 \mathbf{d} & \circlearrowleft_3 \mathbf{d} & \circlearrowleft_1 \mathbf{d} \\ \mathbf{d} & \circlearrowleft_3 \mathbf{d} & \circlearrowleft_1 \mathbf{d} & \circlearrowleft_2 \mathbf{d} \end{pmatrix}, \tag{17}$$

where $\circlearrowleft_i,\ 1 \leq i \leq 3$ are the level permuted designs of the codes $\{0, 1, 2, 3\}$ as given in Table 32 (source [127]). When $m = 1$, [128] studied the one-time quadrupling technique.

Table 32: The selected permutations among the codes $\{0, 1, 2, 3\}$. Source [127]

Code	Level Permutation	Image	Code	Level Permutation	Image
0	$\circlearrowleft_1$	1	1	$\circlearrowleft_1$	0
0	$\circlearrowleft_2$	2	1	$\circlearrowleft_2$	3
0	$\circlearrowleft_3$	3	1	$\circlearrowleft_3$	2
2	$\circlearrowleft_1$	3	3	$\circlearrowleft_1$	2
2	$\circlearrowleft_2$	0	3	$\circlearrowleft_2$	1
2	$\circlearrowleft_3$	1	3	$\circlearrowleft_3$	0

Example 5.3 *[127] Quadrupling of a very simple four-level design with one factor $\mathbf{d} = (0\ 1\ 2\ 3)^\top$ two times by the above*

technique gives the following orthogonal four-level design $\boldsymbol{Q}^{\otimes^2} \in \mathbb{U}_{64}(4^{16})$ *with 16 factors and 64 runs.*

$$\boldsymbol{Q}^{\otimes^2} = \begin{pmatrix} 0123012301230123012301230123012301230123012301230123012301230123 \\ 0123103223013210012310322301321001231032230132100123103223013210 \\ 0123230132101032012323013210103201232301321010320123230132101032 \\ 0123321010322301012332101032230101233210103223010123321010322301 \\ 0123012301230123103210321032103223012301230123013210321032103210 \\ 0123103223013210103201233210230123013210012310323210230110320123 \\ 0123230132101032103232102301012323010123103232103210103201232301 \\ 0123321010322301103223010123321023011032321001233210012323011032 \\ 0123012301230123230123012301230132103210321032101032103210321032 \\ 0123103223013210230132100123103232102301103201231032012332102301 \\ 0123230132101032230101231032321032101032012323011032321023010123 \\ 0123321010322301230110323210012332100123230110321032230101233210 \\ 0123012301230123321032103210321010321032103210322301230123012301 \\ 0123103223013210321023011032012310320123321023012301321001231032 \\ 0123230132101032321010320123230110323210230101232301012310323210 \\ 0123321010322301321001232301103210322301012332102301103232100123 \end{pmatrix}^{\top}.$$

[127] investigated the statistical properties of the generated multiple quadruple designs by the new algorithm, MQA, for constructing optimal four-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs. Furthermore, [129] improved the generated multiple quadruple designs using the level permutations algorithm. That work presents many intricate theoretical relationships between the initial designs and their level-permuted versions, which are used to improve the quality of the resulting four-level designs.

5.4 Adjusted Multiple Doubling Algorithm for Constructing Large Four-Level Designs

[130] extended the idea of the MDA to *Adjusted Multiple Doubling Algorithm (AMQA)* for constructing four-level designs from two-level designs and vice versa. AMDA is a significant and promising technique for constructing optimal four-level designs with large sizes by using very small and simple initial two-level designs as follows in Algorithm 4. [130] investigated the statistical properties of the generated four-level designs by the new algorithm, AMDA, for constructing optimal four-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs.

Algorithm 4: A novel coding scheme from $\mathbb{U}_n(2^s)$ to $\mathbb{U}_{2^m n}(4^{2^{m-1}s})$. Source [130]

- *Step 1*: Given a two-level $\mathbf{d} \in \mathbb{U}_n(2^s)$ extend it using the MDA to generate the double design $\mathbf{D}^{\odot^m} = \odot^m(\mathbf{d}) \in \mathbb{U}_{2^m n}(2^{2^m s})$ as given in (16) [118].

- *Step 2*: Divide the double design $\mathbf{D}^{\odot^m} = \left(\mathbf{d}_r^{\odot^m}\right)_{r=1}^{2^m s} \in \mathbb{U}_{2^m n}(2^{2^m s})$ into $2^{m-1}s$ equal size sub-designs $\mathbf{d}_t^{\odot^m} \in \mathbb{U}_{2^m n}(2^2)$ each sub-design contains the two columns $\mathbf{d}_t^{\odot^m}$ and $\mathbf{d}_{t+2^{m-1}s}^{\odot^m}$ for $1 \le t \le 2^{m-1}s$. That is, construct the following $2^m n \times 2$ sub-designs

$$\mathbf{D}_t^{\odot^m} = \left(\mathbf{d}_t^{\odot^m} \quad \mathbf{d}_{t+2^{m-1}s}^{\odot^m}\right), \ 1 \le t \le 2^{m-1}s. \tag{18}$$

- *Step 3*: Construct the rearranged expanded double design $\mathbf{R}^{\odot^m} \in \mathbb{U}_{2^m n}(2^{2^m s})$ by combining the $2^{m-1}s$ sub-matrices $\mathbf{D}_t^{\odot^m}$, $1 \le t \le 32m - 1s$. That is,

$$\mathbf{R}^{\odot^m} = \left(\mathbf{D}_1^{\odot^m} \ \mathbf{D}_2^{\odot^m} \ \dots \ \mathbf{D}_{2^{m-1}s}^{\odot^m}\right).$$

- *Step 4*: Convert each sub-matrix to a column with four codes as follows

$$F\left(\mathbf{D}_t^{\odot^m}\right) = F\left(\mathbf{d}_t^{\odot^m} \quad \mathbf{d}_{t+2^{m-1}s}^{\odot^m}\right) = \mathbf{Y}_t^{\odot^m}, \tag{19}$$

where $1 \le t \le 2^{m-1}s$ and the map F is defined in Table 33.

- *Step 5*: Combine these $2^{m-1}s$ columns $\mathbf{Y}_t^{\odot^m}$, $1 \le t \le 2^{m-1}s$ to generate the quaternary codes design $\mathbf{Y}^{\odot^m} = \left(\mathbf{Y}_r^{\odot^m}\right)_{r=1}^{2^{m-1}s} \in \mathbb{U}_{2^m n}\left(4^{\zeta} n\right)$. That is,

$$\mathbf{D}^{\odot^m} \in \mathbb{U}_{2^m n}(2^{2^m s}) \implies \mathbf{Y}^{\odot^m} = F\left(\mathbf{D}^{\odot^m}\right) \in \mathbb{U}_{2^m n}(4^{2^{m-1}s}). \tag{20}$$

Table 33: The maps among the binary codes and quaternary codes

Code	Map	Image	Code	Map	Image
(0 0)	$\xrightarrow{F}$	0	(0 1)	$\xrightarrow{F}$	1
(1 0)	$\xrightarrow{F}$	3	(1 1)	$\xrightarrow{F}$	2

Example 5.4 *Using the two-level design* $\mathbf{d} \in \mathbb{U}_2(2^1)$ *in Table 34 as a generator for Algorithm 4. For* $m = 3$*, the expanded binary code design* $\mathbf{D}^{\odot^2} \in \mathbb{U}_{16}(2^{16})$ *and the corresponding generated four-level design* $\mathbf{Y}^{\odot^3} = F\left(\mathbf{D}^{\odot^3}\right) \in \mathbb{U}_{16}(4^8)$ *are given in Table 34 (source [130]).*

Table 34: The kernel two-level design and its corresponding $m(=3)$-double design and $m(=3)$-four-level design using the AMDA. Source [130]

Run No	kernel $\mathbf{d} \in \mathbb{U}_2(2^1)$	$\mathbf{D}^{\odot 3} \in \mathbb{U}_{16}(2^8)$								$\mathbf{Y}^{\odot 3} \in \mathbb{U}_{16}(4^4)$			
		$\mathbf{d}_1^{\odot 3}$	$\mathbf{d}_2^{\odot 3}$	$\mathbf{d}_3^{\odot 3}$	$\mathbf{d}_4^{\odot 3}$	$\mathbf{d}_5^{\odot 3}$	$\mathbf{d}_6^{\odot 3}$	$\mathbf{d}_7^{\odot 3}$	$\mathbf{d}_8^{\odot 3}$	$\mathbf{y}_1^{\odot 3}$ $F(\mathbf{d}_1^{\odot 3}, \mathbf{d}_5^{\odot 3})$	$\mathbf{y}_2^{\odot 3}$ $F(\mathbf{d}_2^{\odot 3}, \mathbf{d}_6^{\odot 3})$	$\mathbf{y}_3^{\odot 3}$ $F(\mathbf{d}_3^{\odot 3}, \mathbf{d}_7^{\odot 3})$	$\mathbf{y}_4^{\odot 3}$ $F(\mathbf{d}_4^{\odot 3}, \mathbf{d}_8^{\odot 3})$
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	2	2	2	2
3		0	1	0	1	0	1	0	1	0	2	0	2
4		1	0	1	0	1	0	1	0	2	0	2	0
5		0	0	1	1	0	0	1	1	0	0	2	2
6		1	1	0	0	1	1	0	0	2	2	0	0
7		0	1	1	0	0	1	1	0	0	2	2	0
8		1	0	0	1	1	0	0	1	2	0	0	2
9		0	0	0	0	1	1	1	1	1	1	1	1
10		1	1	1	1	0	0	0	0	3	3	3	3
11		0	1	0	1	1	0	1	0	1	3	1	3
12		1	0	1	0	0	1	0	1	3	1	3	1
13		0	0	1	1	1	1	0	0	1	1	3	3
14		1	1	0	0	0	0	1	1	3	3	1	1
15		0	1	1	0	1	0	0	1	1	3	3	1
16		1	0	0	1	0	1	1	0	3	1	1	3

5.5 Adjusted Multiple Tripling Algorithm for Constructing Large Nine-Level Designs

[131] extended the idea of the MTA to *Adjusted Multiple Tripling Algorithm (AMTA)* for constructing nine-level designs from three-level designs. The AMTA is a significant and promising technique for constructing optimal nine-level designs with large sizes by using very small and simple initial three-level designs as follows in Algorithm 5 (source [131]). [131] investigated the statistical properties of the generated nine-level designs by the new algorithm, AMTA, for constructing optimal nine-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs.

Algorithm 5: A novel coding scheme from $\mathbb{U}_n(3^s)$ to $\mathbb{U}_{3^m n}(9^{3^{m-1}s})$. Source [131]

- *Step 1*: Given a three-level $\mathbf{d} \in \mathbb{U}_n(3^s)$ extend it using the MTA to generate the triple design $\mathbf{T}^{\uplus^m} = \otimes^m(\mathbf{d}) \in \mathbb{U}_{3^m n}(3^{3^m s})$ as given in (17) [124].
- *Step 2*: Divide the triple design $\mathbf{T}^{\uplus^m} = \left(\mathbf{t}_r^{\uplus^m}\right)_{r=1}^{4^m s} \in \mathbb{U}_{3^m n}(3^{3^m s})$ into $3^{m-1}s$ equal size sub-designs $\mathbf{t}_t^{\uplus^m} \in \mathbb{U}_{3^m n}(3^3)$ each sub-design contains the four columns $\mathbf{t}_t^{\uplus^m}$, $\mathbf{t}_{t+3^{m-1}s}^{\uplus^m}$, and $\mathbf{t}_{t+2\times 3^{m-1}s}^{\otimes^m}$ for $1 \leq t \leq 3^{m-1}s$. That is, construct the following $3^m n \times 3$ sub-designs

$$\mathbf{T}_t^{\uplus^m} = \left(\mathbf{t}_t^{\uplus^m} \;\; \mathbf{t}_{t+3^{m-1}s}^{\uplus^m} \;\; \mathbf{t}_{t+2\times 3^{m-1}s}^{\otimes^s}\right), \; 1 \leq t \leq 3^{m-1}s. \tag{21}$$

- *Step 3*: Construct the rearranged expanded triple design $\mathbf{R}^{\uplus^m} \in \mathbb{U}_{3^m n}(3^{3^m s})$ by combining the $3^{m-1}s$ sub-matrices $\mathbf{T}_t^{\uplus^m}$, $1 \leq t \leq 3^{m-1}s$. That is,

$$\mathbf{R}^{\uplus^m} = \left(\mathbf{T}_1^{\uplus^m} \;\; \mathbf{T}_2^{\uplus^m} \;\; \dots \;\; \mathbf{T}_{3^{m-1}s}^{\uplus^m}\right).$$

- *Step 4*: Convert each sub-matrix to a column with nine codes as follows

$$F\left(\mathbf{T}_t^{\uplus^m}\right) = F\left(\mathbf{t}_t^{\uplus^m} \quad \mathbf{t}_{t+3^{m-1}s}^{\uplus^m} \quad \mathbf{t}_{t+2\times 3^{m-1}s}^{\uplus^\zeta}\right) = \mathbf{Z}_t^{\uplus^m}, \tag{22}$$

where $1 \le t \le 3^{m-1}s$ and the map F is defined in Table 35 (source [131]).

Table 35: The maps among the three codes and nine codes. Source [131]

Code	Map	Image	Code	Map	Image	Code	Map	Image
(0 0 0)	$\xrightarrow{F}$	0	(0 1 1)	$\xrightarrow{F}$	2	(0 2 2)	$\xrightarrow{F}$	1
(1 0 1)	$\xrightarrow{F}$	4	(1 1 2)	$\xrightarrow{F}$	3	(1 2 0)	$\xrightarrow{F}$	5
(2 0 2)	$\xrightarrow{F}$	8	(2 1 0)	$\xrightarrow{F}$	7	(2 2 1)	$\xrightarrow{F}$	6

- *Step 5*: Combine these $3^{m-1}s$ columns $\mathbf{Z}_t^{\uplus^m}$, $1 \le t \le 3^{m-1}s$ to generate the nine codes matrix $\mathbf{Z}^{\uplus^m} = \left(\mathbf{K}_r^{\otimes^m}\right)_{r=1}^{4^{m-1}s} \in \mathbb{U}_{3^m n}\left(9^{3^{m-1}s}\right)$. That is,

$$\mathbf{T}^{\uplus^m} \in \mathbb{U}_{3^m n}(3^{3^m s}) \implies \mathbf{Z}^{\uplus^m} = F\left(\mathbf{T}^{\uplus^m}\right) \in \mathbb{U}_{3^m n}(9^{3^{m-1}s}). \tag{23}$$

Example 5.5 *For the kernel three-level design* $\mathbf{d} = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}^\top \in \mathbb{U}_3(3^2)$, *its corresponding 2-triple design* $\mathbf{T}^{\uplus^2} \in \mathbb{U}_{27}(3^{18})$ *and image nine-level design* $\mathbf{Z}^{\uplus^2} \in \mathbb{U}_{27}(3^6)$ *are given in Table 36 (source [131]) using the above-mentioned AMTA.*

Table 36: The triple design $\mathbf{T}^{\uplus^2} \in \mathbb{U}_{27}(3^{18})$ and image nine-level design $\mathbf{Z}^{\uplus^2} \in \mathbb{U}_{27}(3^6)$ for the example 5.5. Source [131]

$\mathbf{t}_1^{\uplus^2}$	$\mathbf{t}_2^{\uplus^2}$	$\mathbf{t}_3^{\uplus^2}$	$\mathbf{t}_4^{\uplus^2}$	$\mathbf{t}_5^{\uplus^2}$	$\mathbf{t}_6^{\uplus^2}$	$\mathbf{t}_7^{\uplus^2}$	$\mathbf{t}_8^{\uplus^2}$	$\mathbf{t}_9^{\uplus^2}$	$\mathbf{t}_{10}^{\uplus^2}$	$\mathbf{t}_{11}^{\uplus^2}$	$\mathbf{t}_{12}^{\uplus^2}$	$\mathbf{t}_{13}^{\uplus^2}$	$\mathbf{t}_{14}^{\uplus^2}$	$\mathbf{t}_{15}^{\uplus^2}$	$\mathbf{t}_{16}^{\uplus^2}$	$\mathbf{t}_{17}^{\uplus^2}$	$\mathbf{t}_{18}^{\uplus^2}$
0	1	0	1	0	2	0	1	0	1	0	2	0	2	0	2	0	1
1	2	1	2	2	1	1	2	1	2	2	1	2	1	2	1	1	2
2	0	2	0	1	0	2	0	2	0	1	0	1	0	1	0	2	0
0	1	2	0	2	1	0	1	2	0	2	1	0	2	1	0	1	2
1	2	0	1	1	0	1	2	0	1	1	0	2	1	0	2	2	0
2	0	1	2	0	2	2	0	1	2	0	2	1	0	2	1	0	1
0	1	1	2	1	0	0	1	1	2	1	0	0	2	2	1	2	0
1	2	2	0	0	2	1	2	2	0	0	2	2	1	1	0	0	1
2	0	0	1	2	1	2	0	0	1	2	1	1	0	0	2	1	2
0	1	0	1	0	2	2	0	2	0	2	1	2	1	2	1	2	0
1	2	1	2	2	1	0	1	0	1	1	0	1	0	1	0	0	1
2	0	2	0	1	0	1	2	1	2	0	2	0	2	0	2	1	2
0	1	2	0	2	1	2	0	1	2	1	0	2	1	0	2	0	1
1	2	0	1	1	0	0	1	2	0	0	2	1	0	2	1	1	2
2	0	1	2	0	2	1	2	0	1	2	1	0	2	1	0	2	0
0	1	1	2	1	0	2	0	0	1	0	2	2	1	1	0	1	2
1	2	2	0	0	2	0	1	1	2	2	1	1	0	0	2	2	0
2	0	0	1	2	1	1	2	2	0	1	0	0	2	2	1	0	1
0	1	0	1	0	2	1	2	1	2	1	0	1	0	1	0	1	2
1	2	1	2	2	1	2	0	2	0	0	2	0	2	0	2	2	0
2	0	2	0	1	0	0	1	0	1	2	1	2	1	2	1	0	1
0	1	2	0	2	1	1	2	0	1	0	2	1	0	2	1	2	0
1	2	0	1	1	0	2	0	1	2	2	1	0	2	1	0	0	1
2	0	1	2	0	2	0	1	2	0	1	0	2	1	0	2	1	2
0	1	1	2	1	0	1	2	2	0	2	1	1	0	0	2	0	1
1	2	2	0	0	2	2	0	0	1	1	0	0	2	2	1	1	2
2	0	0	1	2	1	0	1	1	2	0	2	2	1	1	0	2	0

$f(\mathbf{t}_1^{\uplus^2}, \mathbf{t}_7^{\uplus^2}, \mathbf{t}_{13}^{\uplus^2})$ $\mathbf{z}_1^{\otimes^2}$	$f(\mathbf{t}_2^{\uplus^2}, \mathbf{t}_8^{\uplus^2}, \mathbf{t}_{14}^{\uplus^2})$ $\mathbf{z}_2^{\otimes^2}$	$f(\mathbf{t}_3^{\uplus^2}, \mathbf{t}_9^{\uplus^2}, \mathbf{t}_{15}^{\uplus^2})$ $\mathbf{z}_3^{\otimes^2}$	$f(\mathbf{t}_4^{\uplus^2}, \mathbf{t}_{10}^{\uplus^2}, \mathbf{t}_{16}^{\uplus^2})$ $\mathbf{z}_4^{\otimes^2}$	$f(\mathbf{t}_5^{\uplus^2}, \mathbf{t}_{11}^{\uplus^2}, \mathbf{t}_{17}^{\uplus^2})$ $\mathbf{z}_5^{\otimes^2}$	$f(\mathbf{t}_6^{\uplus^2}, \mathbf{t}_{12}^{\uplus^2}, \mathbf{t}_{18}^{\uplus^2})$ $\mathbf{z}_6^{\otimes^2}$
0	3	0	3	0	6
3	6	3	6	6	3
6	0	6	0	3	0
0	3	6	0	6	3
3	6	0	3	3	0
6	0	3	6	0	6
0	3	3	6	3	0
3	6	6	0	0	6
6	0	0	3	6	3
1	4	1	4	1	7
4	7	4	7	7	4
7	1	7	1	4	1
1	4	7	1	7	4
4	7	1	4	4	1
7	1	4	7	1	7
1	4	4	7	4	1
4	7	7	1	1	7
7	1	1	4	7	4
2	5	2	5	2	8
5	8	5	8	8	5
8	2	8	2	5	2
2	5	8	2	8	5
5	8	2	5	5	2
8	2	5	8	2	8
2	5	5	8	5	2
5	8	8	2	2	8
8	2	2	5	8	5

5.6 Adjusted Multiple Quadrupling Algorithm for Constructing Large Sixteen-Level Designs

[132] extended the idea of the MQA [127] for four-level designs $\mathbf{d} \in \mathbb{U}_n(4^s)$ to *Adjusted Multiple Quadrupling Algorithm (AMQA)* for constructing sixteen-level designs. AMQA transfers a kernel four-level design $\mathbf{d} \in \mathbb{U}_n(4^s)$ to a sixteen-level FrFD $\mathbf{K}^{\otimes^m} = f(\otimes^m(\mathbf{d})) \in \mathbb{U}_{4^m n}(16^{4^{m-1}s})$ for any $m \geq 1$ as given in Algorithm 6 (source [132]). [132] investigated the statistical properties of the generated sixteen-level designs by the new algorithm, AQTA, for constructing optimal sixteen-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs.

Algorithm 6: A novel coding scheme from $\mathbb{U}_n(4^s)$ to $\mathbb{U}_{4^m n}(16^{4^{m-1}s})$. Source [132]

- *Step 1*: Given a four-level $\mathbf{d} \in \mathbb{U}_n(4^s)$ extend it using the MQA to generate the quadruple design $\mathbf{Q}^{\otimes^m} = \otimes^m(\mathbf{d}) \in \mathbb{U}_{4^m n}(4^{4^m s})$ as given in (18) [127]
- *Step 2*: Divide the quadruple design $\mathbf{Q}^{\otimes^m} = \left(\mathbf{q}_r^{\otimes^m}\right)_{r=1}^{4^m s} \in \mathbb{U}_{4^m n}(4^{4^m s})$ into $4^{m-1}s$ equal size sub-designs $\mathbf{q}_t^{\otimes^m} \in \mathbb{U}_{4^m n}(4^4)$ each sub-design contains the four columns $\mathbf{q}_t^{\otimes^m}$, $\mathbf{q}_{t+4^{m-1}s}^{\otimes^m}$, $\mathbf{q}_{t+2\times 4^{m-1}s}^{\otimes^m}$ and $\mathbf{q}_{t+3\times 4^{m-1}s}^{\otimes^m}$ for $1 \leq t \leq 4^{m-1}s$. That is, construct the following $4^m n \times 4$ sub-designs

$$\mathbf{Q}_t^{\otimes^m} = \left(\mathbf{q}_t^{\otimes^m} \;\; \mathbf{q}_{t+4^{m-1}s}^{\otimes^m} \;\; \mathbf{q}_{t+2\times 4^{m-1}s}^{\otimes^s} \;\; \mathbf{q}_{t+3\times 4^{m-1}s}^{\otimes^m}\right), \; 1 \leq t \leq 4^{m-1}s. \tag{24}$$

- *Step 3*: Construct the rearranged expanded quaternary codes design $\mathbf{R}^{\otimes^m} \in \mathbb{U}_{4^m n}(4^{4^m s})$ by combining the $4^{m-1}s$ sub-matrices $\mathbf{Q}_t^{\otimes^m}$, $1 \leq t \leq 4^{m-1}s$. That is,

$$\mathbf{R}^{\otimes^m} = \left(\mathbf{Q}_1^{\otimes^m} \;\; \mathbf{Q}_2^{\otimes^m} \;\; \ldots \;\; \mathbf{Q}_{4^{m-1}s}^{\otimes^m}\right).$$

- *Step 4*: Convert each sub-matrix to a column with sixteen codes as follows

$$F\left(\mathbf{Q}_t^{\otimes^m}\right) = F\left(\mathbf{q}_t^{\otimes^m} \;\; \mathbf{q}_{t+4^{m-1}s}^{\otimes^m} \;\; \mathbf{q}_{t+2\times 4^{m-1}s}^{\uplus^\zeta} \;\; \mathbf{q}_{t+3\times 4^{m-1}s}^{\uplus^\zeta}\right) = \mathbf{K}_t^{\otimes^m}, \tag{25}$$

where $1 \leq t \leq 4^{m-1}s$ and the map F is defined in Table 37 (source [132]).

Table 37: The maps among the quaternary codes and sixteen codes. Source [132]

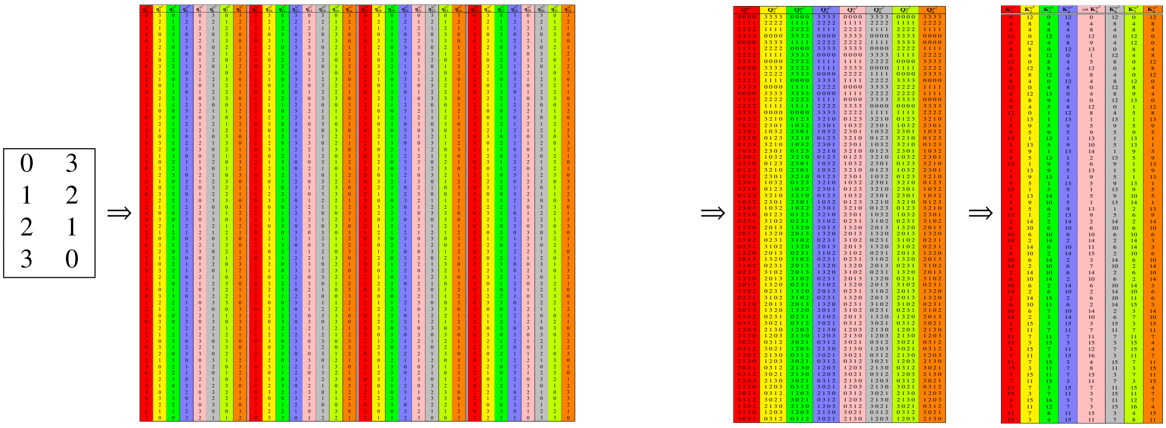
Code	Map	Image	Code	Map	Image	Code	Map	Image	Code	Map	Image
(0 0 0 0)	$\xrightarrow{F}$	0	(1 1 1 1)	$\xrightarrow{F}$	4	(2 2 2 2)	$\xrightarrow{F}$	8	(3 3 3 3)	$\xrightarrow{F}$	12
(0 1 2 3)	$\xrightarrow{F}$	1	(1 0 3 2)	$\xrightarrow{F}$	5	(2 3 0 1)	$\xrightarrow{F}$	9	(3 2 1 0)	$\xrightarrow{F}$	13
(0 2 3 1)	$\xrightarrow{F}$	2	(1 3 2 0)	$\xrightarrow{F}$	6	(2 0 1 3)	$\xrightarrow{F}$	10	(3 1 0 2)	$\xrightarrow{F}$	14
(0 3 1 2)	$\xrightarrow{F}$	3	(1 2 0 3)	$\xrightarrow{F}$	7	(2 1 3 0)	$\xrightarrow{F}$	11	(3 0 2 1)	$\xrightarrow{F}$	15

- *Step 5*: Combine these $4^{m-1}s$ columns $\mathbf{K}_t^{\otimes^m}$, $1 \leq t \leq 4^{m-1}\omega$ to generate the sixteen codes matrix $\mathbf{K}^{\otimes^m} = \left(\mathbf{K}_r^{\otimes^m}\right)_{r=1}^{4^{m-1}s} \in \mathbb{U}_{4^m n}(16^{4^{m-1}s})$. That is,

$$\mathbf{Q}^{\otimes^m} \in \mathbb{U}_{4^m n}(4^{4^m s}) \implies \mathbf{K}^{\otimes^m} = F\left(\mathbf{Q}^{\otimes^m}\right) \in \mathbb{U}_{4^m n}(16^{4^{m-1}s}). \tag{26}$$

Example 5.6 *[132]Using the four-level design $\mathbf{d} \in \mathbb{U}_4(4^2)$ in Table 38 (source [132]) as a generator for Algorithm 6. For $m = 2$, the expanded quaternary design $\mathbf{Q}^{\otimes^2} \in \mathbb{U}_{64}(4^{32})$, the rearranged expanded quaternary design $\mathbf{R}^{\otimes^2} \in \mathbb{U}_{64}(4^{32})$ and the corresponding generated sixteen-level design $\mathbf{K}^{\otimes^2} = F\left(\mathbf{Q}^{\otimes^2}\right) \in \mathbb{U}_{64}(16^8)$ are given in Table 38 (source [132]).*

Table 38: The generator four-level design $\mathbf{d} \in \mathbb{U}_4(4^2)$, the multiple quaternary design $\mathbf{Q}^{\otimes^2} \in \mathbb{Z}_{64}(4^{32})$ and its rearranged matrix, and the final matrix $\mathbf{K}^{\otimes^2} \in \mathbb{Z}_{64}(16^8)$ for the Example 5.6. Source [132]



5.7 Integrating the Algorithms for Constructing Large Mixed-Level Designs

[133] modified the AMDA to transfer a kernel two-level design $\mathbf{d} \in \mathbb{U}_n(2^s)$ to a mixed two- and four-level design from $\mathbb{U}_{2^m n}(2^{2^m s_1} 4^{2^{m-1} s_2})$ for any $m \geq 1,\ s_1 + s_2 = s$ as follows in Algorithm 7 (source [133]). [133] investigated the statistical properties of the generated mixed two- and four-level designs by the new algorithm for constructing optimal mixed two- and four-level designs in view of the above-mentioned four optimization perspective and the results show that the new technique has a good performance for constructing uniform designs.

Algorithm 7: Constructing large three- and nine-level designs. Source [133]

- *Step 1*: For a given kernel two-level design $\mathbf{d} \in \mathbb{U}_n\left(2^s\right)$, divide it into two symmetric sub-designs $\mathbf{d}_\zeta \in \mathbb{U}_n\left(2^{s_\zeta}\right)$, $\zeta = 1, 2$ and $s = s_1 + s_2$. That is, $\mathbf{d} = (\mathbf{d}_1 \ \ \mathbf{d}_2)$.
- *Step 2*: Extend the kernel two-level design $\mathbf{G} \in \mathbb{U}_n\left(2^s\right)$ to an m-double design $\mathbf{D}^{\uplus^m} \in \mathbb{U}_{2^m n}\left(2^{2^m s}\right)$ using a modified version of the AM DA as follows:
$$\mathbf{D}^{\uplus^m} = \left(\mathbf{D}_1^{\uplus^m} \ \ \mathbf{D}_2^{\uplus^m}\right), \tag{27}$$
where $\mathbf{D}_\zeta^{\uplus^m} \in \mathbb{U}_{2^m n}\left(2^{2^m s_\zeta}\right)$ is the multiple double design for the designs $\mathbf{d}_\zeta \in \mathbb{U}_n\left(2^{s_\zeta}\right)$, $\zeta = 1, 2$ using the MDA [76]
- *Step 3*: Transfer the multiple double design $\mathbf{D}_2^{\uplus^m} \in \mathbb{U}_{2^m n}\left(2^{2^m s_2}\right)$ to a four-level design $\mathbf{Z}_2^{\uplus^m} = F(\mathbf{D}_2^{\uplus^m}) \in \mathbb{U}_{2^m n}\left(4^{2^{m-1} s_2}\right)$ using the AMDA [130].
- *Step 4*: The resulting mixed two- and four-level design is given by combining the sub-design with two-level factors $\mathbf{D}_1^{\uplus^m} \in \mathbb{U}_{2^m n}\left(2^{2^m s_1}\right)$ with the sub-design with four-level factors $\mathbf{Z}_2^{\uplus^m} \in \mathbb{U}_{2^m n}\left(4^{2^{m-1} s_2}\right)$ as follows
$$\mathbf{G}^{\uplus^m} = \left(\mathbf{D}_1^{\uplus^m} \ \ \mathbf{Z}_2^{\uplus^m}\right). \tag{28}$$

---

[134] modified the AMTA to transfer a kernel three-level design $\mathbf{d} \in \mathbb{U}_n(3^s)$ to a mixed three- and nine-level design from $\mathbb{U}_{3^m n}(3^{3^{m-1} s_1} 9^{3^{m-1} s_2})$ for any $m \geq 1,\ s_1 + s_2 = s$ as follows in Algorithm 8 (source [134]). [134] investigated the statistical properties of the generated mixed three- and nine-level designs by the new algorithm for constructing optimal mixed three- and nine-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs.

Algorithm 8: Constructing large three- and nine-level designs. Source [134]

- *Step 1*: For a given kernel three-level design $\mathbf{d} \in \mathbb{U}_n\left(3^s\right)$, divide it into two symmetric sub-designs $\mathbf{d}_\zeta \in \mathbb{U}_n\left(3^{s_\zeta}\right)$, $\zeta = 1, 2$ and $s = s_1 + s_2$. That is, $\mathbf{d} = (\mathbf{d}_1 \;\; \mathbf{d}_2)$.
- *Step 2*: Extend the kernel three-level design $\mathbf{G} \in \mathbb{U}_n\left(3^s\right)$ to an m-triple design $\mathbf{T}^{\uplus^m} \in \mathbb{U}_{3^m n}\left(3^{3^m s}\right)$ using a modified version of the AMTA as follows:

$$\mathbf{T}^{\uplus^m} = \left(\mathbf{T}_1^{\uplus^m} \;\; \mathbf{T}_2^{\uplus^m}\right), \tag{29}$$

  where $\mathbf{T}_\zeta^{\uplus^m} \in \mathbb{U}_{3^m n}\left(3^{3^m s_\zeta}\right)$ is the multiple triple design for the designs $\mathbf{d}_\zeta \in \mathbb{U}_n\left(3^{s_\zeta}\right)$, $\zeta = 1, 2$ using the MTA .
- *Step 3*: Transfer the multiple triple design $\mathbf{T}_2^{\uplus^m} \in \mathbb{U}_{3^m n}\left(3^{3^m s_2}\right)$ to a nine-level design $\mathbf{Z}_2^{\uplus^m} = F(\mathbf{T}_2^{\uplus^m}) \in \mathbb{U}_{3^m n}\left(9^{3^{m-1} s_2}\right)$ using the AMTA.
- *Step 4*: The resulting mixed three- and nine-level design is given by combining the sub-design with three-level factors $\mathbf{T}_1^{\uplus^m} \in \mathbb{U}_{3^m n}\left(3^{3^m s_1}\right)$ with the sub-design with nine-level factors $\mathbf{Z}_2^{\uplus^m} \in \mathbb{U}_{3^m n}\left(9^{3^{m-1} s_2}\right)$ as follows

$$\mathbf{G}^{\uplus^m} = \left(\mathbf{T}_1^{\uplus^m} \;\; \mathbf{Z}_2^{\uplus^m}\right). \tag{30}$$

---

[135] successfully combined the MDA and the MTA for extending a small kernel mixed two-and three-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 3^{s_2})$ to a large mixed two-and three-level design $\mathbf{E}^{\odot\uplus m} \in \mathbb{U}_{3^m n}(2^{4^m s_1} 3^{3^m s_2})$ as in Algorithm 9 (source [135]). [135] investigated the statistical properties of the generated mixed two- and three-level designs by the new algorithm for constructing optimal large mixed two- and three-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs. Moreover, [136] extended the work of [135] for extending a small kernel mixed two-and three-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 3^{s_2})$ to a large mixed two-, three- and nine-level design $\mathbf{E}^{\odot\uplus m} \in \mathbb{U}_{3^m n}(2^{4^m s_1} 3^{3^m s_2} 9^{3^{m-1} s_3})$.

Algorithm 9: A novel coding scheme from $\mathbb{U}_n(2^{s_1} 3^{s_2})$ to $\mathbb{U}_{3^m n}(2^{4^m s_1} 3^{3^m s_2})$. Source [135]

- *Step 1*: For a small kernel mixed two-and three-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 3^{s_2})$, divide it into two symmetric sub-designs $\mathbf{d}_1 \in \mathbb{U}_n(2^s)$ and $\mathbf{d}_2 \in \mathbb{U}_n(3^{s_2})$, where $\mathbf{d}_1 \subset \mathbf{d}$ is the design of all the s_1 representative columns of the two-level factors in the kernel design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 3^{s_2})$, and $\mathbf{d}_2 \subset \mathbf{d}$ is the design of all the s_2 representative columns of the three-level factors in the kernel design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 3^{s_2})$. That is, $\mathbf{d} = (\mathbf{d}_1 \; \mathbf{d}_2)$.
- *Step 2*: Extend the kernel design $\mathbf{d} = (\mathbf{d}_1 \; \mathbf{d}_2) \in \mathbb{U}_n(2^{s_1} 3^{s_2})$ to a large design with $3^m n$ runs and $4^m s_1$ two-level factors and $3^m s_2$ three-level factors $\mathbf{E}^{\odot\uplus m} \in \mathbb{U}_{3^m n}(2^{4^m s_1} 3^{3^m s_2})$ using a modified MDA and MTA as follows:

$$\mathbf{E}^{\odot\uplus m} = \left(\mathbf{D}^{\odot m} \; \mathbf{T}^{\uplus m}\right),$$

  where

$$\mathbf{E}^{\odot\uplus m} = \begin{pmatrix} \mathbf{D}^{\odot\eta-1} & \mathbf{I}_{3^{\eta-1}n}^{4^{\eta-1}s} - \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{I}_{3^{\eta-1}n}^{4^{\eta-1}s} - \mathbf{D}^{\odot\eta-1} & \mathbf{T}^{\uplus^{\eta-1}} & \mathbf{T}^{\uplus^{\eta-1}} & \circlearrowleft_1 \mathbf{T}^{\uplus^{\eta-1}} \\ \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{T}^{\uplus^{\eta-1}} & \circlearrowleft_4 \mathbf{T}^{\uplus^{\eta-1}} & \circlearrowleft_2 \mathbf{T}^{\uplus^{\eta-1}} \\ \mathbf{D}^{\odot\eta-1} & \mathbf{I}_{3^{\eta-1}n}^{4^{\eta-1}s} - \mathbf{D}^{\odot\eta-1} & \mathbf{I}_{3^{\eta-1}n}^{4^{\eta-1}s} - \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{T}^{\uplus^{\eta-1}} & \circlearrowleft_5 \mathbf{T}^{\uplus^{\eta-1}} & \circlearrowleft_3 \mathbf{T}^{\uplus^{\eta-1}} \end{pmatrix},$$

  $1 \leq \eta \leq m$, $\mathbf{I}_{3^{\eta-1}n}^{4^{\eta-1}s}$ is a matrix of ones with $3^{\eta-1}n$ runs and $4^{\eta-1}s$ columns and $\circlearrowleft_k$, $k = 1, 2, 3, 4, 5$ are the level permutations of the three levels that are given in Table 29.

---

[137] successfully combined the MDA and the MQA for extending a small kernel mixed two-and four-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1} 4^{s_2})$ to a large mixed two-and four-level design $\mathbf{E}^{\odot\otimes m} \in \mathbb{U}_{4^m n}(2^{4^m s_1} 4^{4^m s_2})$ as given in Algorithm 10 (source [137]). [137] investigated the statistical properties of the generated mixed two- and three-level designs by the new algorithm for constructing optimal large mixed two- and four-level designs in view of the above-mentioned four optimization perspectives and the results show that the new technique has a good performance for constructing uniform designs. Moreover, [138] extended the work of [137] for extending a small kernel four-level design $\mathbf{d} \in \mathbb{U}_n(4^s)$ to a large mixed four- and sixteen-level design by an algorithm similar to the above algorithms with the obvious changes. The results show that the new technique has a good performance for constructing uniform designs.

Algorithm 10: A novel coding scheme from $\mathbb{U}_n(2^{s_1}4^{s_2})$ to $\mathbb{U}_{4^m n}(2^{4^m s_1}4^{4^m s_2})$. Source [137]

- *Step 1*: For a small kernel mixed two-and four-level design $\mathbf{d} \in \mathbb{U}_n(2^{s_1}4^{s_2})$, divide it into two symmetric sub-designs $\mathbf{d}_1 \in \mathbb{U}_n(4^{s})$ and $\mathbf{d}_2 \in \mathbb{U}_n(4^{s_2})$, where $\mathbf{d}_1 \subset \mathbf{d}$ is the design of all the s_1 representative columns of the two-level factors in the kernel design $\mathbf{d} \in \mathbb{U}_n(2^{s_1}4^{s_2})$, and $\mathbf{d}_2 \subset \mathbf{d}$ is the design of all the s_2 representative columns of the four-level factors in the kernel design $\mathbf{d} \in \mathbb{U}_n(2^{s_1}4^{s_2})$. That is, $\mathbf{d} = (\mathbf{d}_1\ \mathbf{d}_2)$.
- *Step 2*: Extend the kernel design $\mathbf{d} = (\mathbf{d}_1\ \mathbf{d}_2) \in \mathbb{U}_n(2^{s_1}4^{s_2})$ to a large design with $4^m n$ runs and $4^m s_1$ two-level factors and $4^m s_2$ four-level factors $\mathbf{E}^{\odot\otimes m} \in \mathbb{U}_{4^m n}(2^{4^m s_1}4^{4^m s_2})$ using a modified MDA and MQA as follows:

$$\mathbf{E}^{\odot\otimes m} = \left(\mathbf{D}^{\odot m}\ \mathbf{Q}^{\otimes m}\right),$$

where

$$\mathbf{E}^{\odot\otimes m} = \begin{pmatrix} \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{Q}^{\otimes^{\eta-1}} & \mathbf{Q}^{\otimes^{\eta-1}} & \mathbf{Q}^{\otimes^{\eta-1}} & \mathbf{Q}^{\otimes^{\eta-1}} \\ \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_1 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_2 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_3 \mathbf{Q}^{\otimes^{\eta-1}} \\ \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{3^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_2 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_3 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_1 \mathbf{Q}^{\otimes^{\eta-1}} \\ \mathbf{D}^{\odot\eta-1} & \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n} - \mathbf{D}^{\odot\eta-1} & \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_3 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_1 \mathbf{Q}^{\otimes^{\eta-1}} & \circlearrowleft_2 \mathbf{Q}^{\otimes^{\eta-1}} \end{pmatrix},$$

$1 \le \eta \le m$, $\mathbf{I}^{4^{\eta-1}s}_{4^{\eta-1}n}$ is a matrix of ones with $4^{\eta-1}n$ runs and $4^{\eta-1}s$ columns and $\circlearrowleft_i,\ 1 \le i \le 3$ are the level permuted designs of the codes $\{0, 1, 2, 3\}$ as given in Table 32.

---

[141] successfully combined the MDA and the Foldover Algorithm for constructing constructing asymptotically orthogonal maximin distance Latin hypercube designs as given in Algorithm 11 (source [141]). Recently, [142] proposed seven novel theoretical techniques for constructing orthogonal maximin distance uniform projection designs. The proposed designs demonstrate superior performance as the number of factors increases, making them particularly well-suited for surrogate modeling and linear trend estimation in high-dimensional Gaussian processes. Comparative studies show that the proposed techniques outperform existing methods

---

Algorithm 11: Constructing asymptotically OMDLHDs. Source [141]

- *Step 1:* For an integer $\alpha \ge 2$, generate the following two $2^\alpha \times 2^\alpha$ matrices

$$\mathbf{A}_\alpha = \begin{pmatrix} \mathbf{A}_{\alpha-1} & -\mathbf{A}^\star_{\alpha-1} \\ \mathbf{A}_{\alpha-1} & \mathbf{A}^\star_{\alpha-1} \end{pmatrix} \text{ and } \mathbf{B}_\alpha = \begin{pmatrix} \mathbf{B}_{\alpha-1} & -\mathbf{B}^\star_{\alpha-1} - 2^{\alpha-1}\mathbf{B}^\star_{\alpha-1} \\ \mathbf{B}_{\alpha-1} + 2^{\alpha-1}\mathbf{B}_{\alpha-1} & \mathbf{B}^\star_{\alpha-1} \end{pmatrix},$$

where $\mathbf{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $\mathbf{B}_1 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$ and $\mathbf{M}^\star = \begin{pmatrix} -\mathbf{M}_1 \\ \mathbf{M}_2 \end{pmatrix}$ for any $n \times m$ matrix $\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{pmatrix}$ with even n and $\mathbf{M}_1$ and $\mathbf{M}_2$ are the top and bottom half, respectively.
- *Step 2:* Generate a $2^{\alpha+1} \times 2^\alpha$ matrix with fold-over structure by combining the matrices $\mathbf{A}_\alpha$ and $\mathbf{B}_\alpha$ as follows

$$\mathbf{C}_\alpha = \begin{pmatrix} \mathbf{B}_\alpha - \frac{1}{2}\mathbf{A}_\alpha \\ \frac{1}{2}\mathbf{A}_\alpha - \mathbf{B}_\alpha \end{pmatrix}.$$

- *Step 3:* Generate the following $2^{\alpha+1} \times 2^{\alpha-1}$ matrix

$$\mathbf{H}_\alpha = \begin{pmatrix} \mathbf{U}_{\alpha-1} \\ \mathbf{V}_{\alpha-1} \end{pmatrix},$$

where $\mathbf{U}_{\alpha-1} = (U_{ij})$, $\mathbf{V}_{\alpha-1} = (V_{ij})$, $\mathbf{C}_{\alpha-1} = (C_{ij})$, $\mathbf{S}_{\alpha-1} = (S_{ij})$, $S_{ij} = \begin{cases} +1, \text{ if } C_{ij} \ge 0; \\ -1, \text{ if } C_{ij} < 0 \end{cases}$ and

$$\begin{cases} U_{ij} = S_{ij}(2|C_{ij}| - \frac{1}{2}),\ V_{ij} = S_{ij}(2|C_{ij}| + \frac{1}{2}),\ \text{for } 1 \le i \le 2^{\alpha-1}; \\ U_{ij} = S_{ij}(2|C_{ij}| + \frac{1}{2}),\ V_{ij} = S_{ij}(2|C_{ij}| - \frac{1}{2}),\ \text{for } 2^{\alpha-1} + 1 \le i \le 2^\alpha. \end{cases}$$

- *Step 4:* The newly constructed $LHD(2^{\alpha+1}, 3 \times 2^{\alpha-1})$ is given by column-combining the two matrices $\mathbf{C}_\alpha$ and $\mathbf{H}_\alpha$ as follows $\mathbf{L}_\alpha = (\mathbf{C}_\alpha\ \ \mathbf{H}_\alpha)$.

5.8 Clarifying the Significance of the Selected Contributions

This section provides important discussions to help the reader better understand the value of the selected contributions.

- *From the perspective of experimental design criteria:* Without relying on computer search, the new algorithms construct new recommended optimal designs that are either superior to existing recommended designs or are impossible to construct using existing techniques due to their large size (as exemplified in Section 5.1). Interested readers can find further discussions for various levels and cases in the cited references. The theoretical results in these references provide useful benchmarks for experimenters to improve designs generated via any of the aforementioned algorithms. For instance, as shown in [132], for any design **K** generated by any of the above techniques, if

$$\text{Av}\,[\text{Disc}\,(\mathbf{K})]^2 < [\text{Disc}\,(\mathbf{K})]^2\,,$$

then permuting the sixteen levels of one or more columns of **K** is guaranteed to improve its performance with respect to the discrepancy. That is, there exists a level-permuted design $\mathbf{K}_{\circlearrowleft}$ such that

$$[\text{Disc}\,(\mathbf{K}_{\circlearrowleft})]^2 < [\text{Disc}\,(\mathbf{K})]^2\,.$$

Consequently, the proposed technique can be improved from an experimental design perspective by combining it with the (adjusted) TA algorithm. This combination can be implemented in three basic steps [132]:

  - For a given optimization criterion, use the (adjusted) TA algorithm to generate an optimal kernel design **d**.
  - For a suitable $m > 0$, use the optimal kernel design **K** as a generator for any of the aforementioned algorithms to produce a design **K**.
  - For the same optimization criterion, apply the (adjusted) TA algorithm again to improve the generated design **K**.

Figure 15 (source: [132]) provides further details for these steps using the CDisc as the optimization criterion and the AMQA algorithm [132] as an example. Using high-quality designs as generators is far superior to using randomly generated designs. Since designs generated via the new algorithms are efficient, they can serve as effective generators for iterative stochastic search algorithms. To illustrate, the combination of the new AMQA [132] and the iterative stochastic search TA algorithm from the R package `UniDOE` [143] is used to generate sixteen-level designs **K**. A comparative study is presented between designs generated solely by the iterative stochastic search TA algorithm (denoted $\mathbf{K}_{TA} \in \mathbb{U}_{4^m n}(16^{4^{m-1}s})$ and those generated by combining the new AMQA [132] with the threshold accepting algorithm (denoted $\mathbf{K}_{[132]} \in \mathbb{U}_{4^m n}(16^{4^{m-1}s})$). The designs $\mathbf{K}_{TA}$ and $\mathbf{K}_{[132]}$ and their performance are given in Table 39 (source: [132]). Table 40 (source: [132]) shows that the performance of $\mathbf{K}_{[132]}$ is superior to that of $\mathbf{K}_{TA}$.

- *From the perspective of modeling:* [138] addressed the following logical question: How beneficial is the performance of the m-stage designs constructed via the new algorithms? To answer this question and provide a clear understanding of the algorithms' merits, a comparative study is presented between the modeling performance of the m-stage designs and that of designs generated using existing techniques. The study employs various models (polynomial, spline, and Kriging), variable selection techniques, and criteria (F-test, AIC, BIC, RIC, SCAD, LASSO). Performance is evaluated using multiple error measures: mean square error (MSE), root mean square error (RMSE), normalized RMSE (NRMSE), mean absolute error (MAE), and mean relative error (MRE). [138] uses the challenging four-dimensional Wood function:

$$\begin{aligned} y = {} & 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1), \quad (x_1, x_2, x_3, x_4) \in (0, 1)^4. \end{aligned}$$

Models are fitted using four designs of equal size ($n = 16$), each with factors taking 16 distinct values to ensure a fair comparison: (i) a uniform sixteen-level design generated via the TA algorithm using the R package `UniDOE` [143]; (ii) a Latin hypercube sample (LHS), popular in computer experiments; (iii) a uniform design generated by combining the TA algorithm and the coordinate gradient descent (CGD) algorithm [144]; and (iv) the new m-stage design from [132] using the kernel design

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{pmatrix} \in \mathbb{U}(4, 4^4).$$

Table 40 (source: [138]) presents the datasets and their outputs. A polynomial model is fitted to each dataset using the aforementioned criteria and techniques; the results are shown in Table 41 (source: [138]). Figure 15 (source:

[138]) summarizes these results by highlighting the minimum error values (MSE, RMSE, NRMSE, MAE, MRE) under each criterion. Both Table 41 and Figure 16 (source: [138]) demonstrate that, under the polynomial model, the new design outperforms those from existing algorithms across all criteria. Under the spline model, all four designs exhibit very small error values and minimal differences when using AIC, BIC, RIC, SCAD, and LASSO; however, they show significant error values under the F-test [138]. For further details on the modeling behavior of the new designs, interested readers are referred to the cited references.

- *Isomorphism detection:* Isomorphism examination [139, 140] plays a vital role in design enumeration and optimal design search. Isomorphic designs, which are statistically and mathematically equivalent, can be created by reordering runs, relabeling factors, or switching levels. The theoretical results for $Av[Disc]$ of designs constructed via the above algorithms can serve as benchmarks for detecting non-isomorphic designs: if two kernel designs have different performances under the HDP or GWLP, then designs generated from them via any of the algorithms are non-isomorphic. Thus, two generated designs $\mathbf{S}_1$ and $\mathbf{S}_2$ are non-isomorphic if

$$\text{Av}\left[\text{Disc}\left(\mathbf{S}_1\right)\right]^2 \neq \text{Av}\left[\text{Disc}\left(\mathbf{S}_2\right)\right]^2.$$

  However, two non-isomorphic designs may share the same $Av[Disc]$. In such cases, the full discrepancy distribution of all level-permuted versions of $\mathbf{S}_1$ and $\mathbf{S}_2$ must be examined; if the distributions differ, the designs are non-isomorphic.

Table 39: The generated sixteen codes designs using the new hybrid algorithm and the iterative threshold accepting algorithm. Source [132]

	$\mathbf{K}_{TA}^{\top} \in \mathbb{U}_{16}\left(16^{2}\right)$	$\mathbf{K}_{[132]}^{\top} \in \mathbb{U}_{16}\left(16^{2}\right)$
	2 11 4 10 13 9 7 3 5 14 12 0 8 1 6 15 1 11 15 7 14 0 5 8 3 2 4 6 13 12 10 9	6 2 1 13 14 0 7 9 4 5 8 15 3 11 10 12 13 12 2 7 14 10 3 15 0 8 9 4 6 11 5 1
WDisc	0.0027	0.0026
	$\mathbf{K}_{TA}^{\top} \in \mathbb{U}_{16}\left(16^{4}\right)$	$\mathbf{K}_{[132]}^{\top} \in \mathbb{U}_{16}\left(16^{4}\right)$
	4 15 11 3 10 5 6 1 9 12 7 0 13 8 2 14 8 5 14 13 9 2 10 0 3 7 4 11 1 15 6 12 10 4 12 5 6 2 14 8 11 1 7 0 15 3 13 9 9 5 6 1 12 14 3 4 0 2 7 8 10 11 13 15	13 0 1 5 4 8 10 11 9 12 15 7 3 6 2 14 11 7 13 15 5 10 1 14 6 8 0 3 9 12 2 4 10 4 12 9 0 1 7 3 11 14 15 13 8 5 2 6 5 6 2 15 4 1 3 8 11 14 10 7 9 12 13 0
WDisc	0.0193	0.0184
	$\mathbf{K}_{TA}^{\top} \in \mathbb{U}_{16}\left(16^{8}\right)$	$\mathbf{K}_{[132]}^{\top} \in \mathbb{U}_{16}\left(16^{8}\right)$
	10 13 11 2 15 7 12 4 7 7 3 2 10 6 3 15 4 14 8 12 0 6 6 1 3 6 15 0 13 8 2 3 9 11 10 5 12 10 15 9 7 1 2 0 8 4 13 9 13 11 9 5 1 12 4 14 14 8 5 14 1 11 5 0 5 2 1 0 10 15 9 9 3 3 14 9 11 4 7 14 1 4 8 10 0 10 5 14 13 10 0 7 13 0 4 6 13 14 12 1 5 2 6 3 6 5 15 12 9 4 11 11 6 1 7 8 2 7 13 3 15 12 2 8 11 15 12 8 12 10 15 4 10 0 15 6 6 11 3 0 8 11 11 14 12 8 9 2 13 14 2 5 10 4 5 1 11 7 7 15 7 4 13 0 0 3 7 2 13 13 10 3 12 1 2 14 6 5 4 5 14 9 6 1 12 1 9 3 9 8 15 8 13 2 6 1 7 8 15 15 4 10 6 4 5 1 14 14 6 12 1 1 5 11 5 0 3 13 9 8 9 14 10 9 5 2 10 13 14 3 0 15 2 7 15 10 4 0 7 0 6 11 9 7 12 8 13 3 4 12 8 11 12 11 3 2 8 13 8 5 2 9 9 3 11 4 1 13 14 0 10 4 15 10 11 10 3 15 10 9 7 0 12 14 11 12 1 2 4 15 13 4 3 2 14 15 5 12 14 8 6 8 5 1 7 0 7 6 7 1 11 6 0 2 9 13 6 5 12 3 8 7 2 6 6 5 7 5 12 6 8 10 11 14 0 12 8 3 3 14 14 0 7 1 2 2 1 2 11 10 3 11 0 13 4 13 1 0 5 4 1 13 15 12 13 11 3 9 15 10 9 4 5 12 9 9 4 15 15 14 10 7 6 8 10 12 14 7 11 6 3 8 11 3 13 15 8 13 14 9 5 6 5 10 15 11 2 12 4 3 2 9 13 4 10 4 7 3 1 11 5 0 15 13 9 6 2 5 0 1 1 6 4 9 12 14 0 1 0 8 7 14 8 7 12 15 10 2 9 6 4 3 13 6 0 10 0 4 4 12 4 14 5 5 12 12 9 14 8 2 11 11 0 8 7 9 10 13 6 10 14 9 15 11 3 10 1 7 13 1 15 13 6 5 1 7 8 0 2 15 14 12 3 15 2 1 3 5 7 11 8 2	13 15 2 2 11 6 9 7 2 3 11 13 8 11 5 7 6 12 11 15 0 9 7 8 10 14 7 3 1 3 10 1 15 6 14 14 10 8 15 9 12 6 8 1 10 5 9 4 14 4 4 1 5 0 13 0 12 3 12 0 4 5 2 13 9 10 12 1 2 1 7 3 5 4 1 13 6 11 2 0 7 8 10 5 9 5 10 0 15 4 13 0 3 8 12 15 7 4 12 2 8 9 13 11 4 13 15 8 3 5 14 3 6 11 14 7 6 11 14 2 6 12 15 0 9 10 14 1 13 1 0 9 14 7 11 6 5 10 15 5 1 5 3 13 0 2 3 3 14 6 12 6 1 12 10 2 0 7 9 11 14 4 3 0 11 2 7 15 8 14 8 8 13 15 4 11 7 8 1 4 12 6 12 10 9 15 9 4 5 10 13 2 2 2 12 2 14 11 12 10 15 14 5 8 9 4 1 15 14 3 11 6 6 2 8 3 1 1 4 6 4 8 9 11 10 7 15 8 0 13 13 7 12 3 5 10 9 11 0 3 5 1 10 4 5 14 13 0 15 0 7 9 13 6 7 12 14 3 13 8 5 14 3 4 6 0 0 14 5 13 8 12 7 5 8 15 9 12 15 9 15 4 6 4 11 12 1 5 12 10 11 2 10 13 7 6 9 2 3 15 8 1 0 14 7 4 10 1 11 1 10 13 2 9 11 6 3 7 2 0 15 11 10 0 11 3 0 13 6 5 14 0 5 5 7 1 13 1 12 1 7 7 6 11 8 2 14 10 0 13 10 7 8 10 14 3 3 2 5 9 14 6 3 11 4 15 12 9 8 15 2 4 12 1 9 12 9 4 4 15 8 2 13 6 0 2 15 13 14 7 6 0 5 2 5 3 13 11 1 1 10 7 8 14 12 9 8 4 5 12 12 9 8 14 1 3 4 6 11 4 15 3 6 2 2 15 10 10 9 11 13 11 0 4 0 1 3 9 10 6 8 7 14 15 13 5 7 12 2 7 13 15 12 3 7 14 5 12 6 12 6 2 7 10 2 13 10 1 12 9 15 1 14 4 9 3 5 5 5 2 4 11 15 9 11 8 14 0 3 3 13 9 0 13 4 8 10 11 1 15 14 6 6 0 1 8 7 8 0 4 10 11
WDisc	0.0836	0.0792
	$\mathbf{K}_{TA}^{\top} \in \mathbb{U}_{64}\left(16^{16}\right)$	$\mathbf{K}_{[132]}^{\top} \in \mathbb{U}_{64}\left(16^{16}\right)$
	1 6 9 1 14 9 2 8 13 5 2 11 3 0 0 11 6 7 1 7 12 9 8 4 8 4 15 3 14 10 0 4 9 11 11 8 14 10 15 4 13 7 7 6 1 2 0 13 5 13 10 6 10 15 15 5 12 2 3 14 12 12 5 3 7 4 3 2 6 7 10 1 7 1 13 9 11 5 6 3 0 3 0 15 1 15 5 11 10 4 2 15 13 2 2 3 14 12 5 10 4 0 11 8 10 12 8 12 8 12 14 1 0 14 5 9 13 6 15 6 14 13 4 9 9 11 7 8 11 4 11 14 7 2 7 2 4 7 10 0 9 13 3 10 12 8 1 5 11 14 5 6 13 0 3 1 13 2 6 11 8 0 13 1 0 6 12 1 9 12 3 3 14 7 4 15 9 2 5 15 14 8 15 10 10 4 9 6 8 5 15 12 12 1 5 1 8 7 14 13 0 11 7 2 6 0 11 15 12 14 5 5 3 2 4 0 9 10 4 8 6 1 9 8 0 15 7 3 5 6 13 14 2 7 6 9 12 1 4 10 2 11 12 14 10 3 15 15 10 13 9 11 8 13 4 3 10 15 9 4 3 6 7 0 9 8 0 11 12 2 12 7 5 11 5 8 6 0 14 9 5 1 7 10 11 11 14 3 5 12 13 3 0 1 15 13 4 6 12 7 8 15 2 10 13 3 2 1 9 8 14 4 13 6 10 15 1 4 14 2 1 8 15 1 7 5 3 11 10 7 10 13 11 14 5 6 11 2 13 14 2 7 2 4 14 3 12 4 6 10 6 8 0 7 3 12 0 1 0 15 3 12 0 1 9 15 6 14 5 2 15 4 4 8 9 13 11 9 12 13 10 8 9 5 2 10 13 14 12 10 7 7 2 1 11 12 8 9 13 9 10 4 3 9 13 3 11 12 4 6 0 5 10 4 15 0 6 7 0 15 1 11 15 14 5 1 3 9 4 1 2 8 5 8 6 2 15 6 12 11 3 13 14 5 14 0 7 8 9 1 3 5 14 3 2 2 0 1 4 2 13 13 5 9 10 4 4 14 12 10 11 6 1 8 10 8 7 14 11 12 5 11 2 9 15 7 3 7 11 6 13 10 12 9 3 8 15 0 6 13 15 5 0 14 4 15 0 12 8 6 7 1 5 8 11 10 14 1 14 4 12 0 12 5 11 1 10 9 13 5 6 7 5 3 15 3 10 8 2 14 4 13 7 4 13 0 13 15 11 0 0 12 2 8 3 10 15 6 15 14 12 6 3 9 7 9 9 2 1 4 2 8 6 11 1 7 3 3 13 8 9 5 3 15 11 9 4 9 7 10 0 5 11 10 7 2 1 8 7 0 1 10 5 13 9 4 11 1 6 1 13 13 0 14 3 8 12 10 12 15 14 15 2 0 2 7 4 5 6 14 14 15 11 12 6 6 2 8 4 12 12 0 2 4 11 7 3 13 5 15 15 3 4 14 14 14 8 5 11 12 13 5 6 11 6 9 3 2 0 10 2 7 15 5 13 14 0 4 8 9 10 1 15 10 0 7 8 7 1 1 2 3 9 6 11 1 9 6 10 13 4 12 12 8 3 14 2 6 12 2 0 12 10 5 6 15 12 1 13 10 3 11 13 5 9 10 5 7 7 2 8 9 3 3 7 15 15 5 8 4 7 13 12 10 14 13 0 11 14 2 11 15 1 4 9 4 14 4 1 6 6 9 0 8 0 1 11 8 11 10 7 6 6 14 15 4 6 12 12 1 11 13 0 3 8 10 8 1 15 14 2 8 4 5 4 2 4 15 14 2 3 8 9 10 13 0 5 14 12 15 7 13 3 3 9 12 5 7 11 0 9 11 13 9 0 5 2 7 10 1 6 1 8 6 9 15 5 5 15 14 1 10 1 12 7 12 7 11 12 4 2 4 1 6 13 12 8 6 7 11 2 9 3 14 7 1 6 2 11 8 4 10 9 13 15 0 4 5 13 3 0 10 0 2 14 14 8 5 15 9 0 13 11 3 10 3 7 14 1 8 11 8 1 8 4 3 15 7 6 14 0 6 12 15 2 9 13 1 3 5 13 4 15 14 2 0 9 2 4 12 9 6 6 12 3 13 9 10 13 2 5 11 10 8 7 11 5 10 15 12 4 0 5 7 10 1 3 14 11 0 1 3 0 15 1 15 11 0 13 10 7 9 10 3 13 7 13 14 5 14 10 6 4 5 8 9 2 15 12 8 8 14 0 0 5 11 12 8 7 2 13 1 7 3 6 12 9 4 2 5 12 4 11 6 14 9 3 4 6 15 2 10 11 1	3 3 8 10 0 11 9 13 5 7 0 4 15 6 6 12 7 13 11 1 6 15 10 3 14 1 2 13 0 7 11 7 2 3 6 4 4 8 9 1 2 5 8 14 1 14 9 10 13 15 11 5 12 4 12 10 15 8 5 2 14 9 12 0 6 4 9 4 11 3 10 0 3 13 1 14 12 12 15 14 8 5 1 0 9 8 5 0 14 7 12 13 3 15 10 6 1 6 5 10 10 8 13 6 11 2 4 3 7 11 1 13 9 9 8 2 7 0 14 2 15 11 15 12 5 2 7 4 11 14 10 4 13 12 3 13 5 12 2 1 9 10 0 4 5 1 6 4 11 9 11 7 10 0 14 5 10 7 8 13 11 3 3 15 6 15 1 4 8 2 15 14 7 3 13 5 2 14 1 8 12 12 15 8 0 6 9 2 7 0 9 6 11 15 3 15 14 2 11 13 13 9 4 7 14 1 12 0 0 0 7 8 6 13 4 9 15 1 2 4 7 2 6 8 0 3 3 8 15 14 7 9 6 14 9 10 2 6 3 10 1 5 10 11 12 1 5 12 12 13 4 11 10 5 8 5 1 6 10 10 4 0 7 15 12 12 1 3 15 9 11 8 1 4 14 0 0 7 13 8 1 13 14 6 10 5 6 4 10 15 8 8 13 12 14 11 2 5 2 4 3 12 14 5 2 5 0 15 7 6 11 3 11 3 2 9 13 7 9 9 13 14 15 7 15 7 9 6 2 14 5 8 10 6 13 0 8 11 13 0 11 8 3 7 0 1 7 12 11 1 13 6 3 14 0 2 4 11 1 6 3 10 2 15 10 8 9 4 2 5 12 4 4 12 10 15 3 5 9 12 9 5 1 14 8 1 7 8 9 9 13 14 15 6 10 11 3 15 2 5 1 14 0 5 13 11 4 10 15 11 7 7 2 0 9 5 13 14 9 3 12 10 8 1 2 6 13 12 5 3 0 12 6 0 1 7 2 4 12 4 8 4 11 15 6 3 14 10 12 3 0 8 9 3 0 11 2 8 13 11 14 10 15 11 8 15 7 1 6 6 12 3 4 7 15 2 8 13 14 5 0 4 5 9 12 1 2 15 4 13 14 12 11 14 9 7 7 1 10 9 2 6 5 0 6 4 1 5 3 10 13 10 14 5 12 15 8 9 6 3 8 6 6 12 4 14 9 10 5 15 10 0 0 10 2 1 12 0 11 0 1 1 3 10 5 13 4 15 9 1 5 3 11 13 11 2 6 12 14 13 3 7 9 4 13 7 4 15 14 2 7 2 8 8 7 11 11 13 10 7 9 13 7 3 9 14 4 11 11 13 4 2 4 12 10 13 8 15 8 10 5 5 6 14 2 7 3 2 1 3 15 4 14 2 6 9 8 8 15 7 14 1 15 1 10 0 0 0 6 6 9 1 12 12 3 0 5 11 12 5 7 10 3 5 14 0 9 3 7 0 7 3 4 10 4 1 14 12 2 10 1 11 11 11 8 2 12 6 13 12 15 8 3 13 4 6 13 7 12 15 5 1 15 5 5 8 6 4 9 2 11 9 0 8 10 13 14 1 15 0 2 14 6 9 12 1 14 9 14 0 13 2 11 9 9 8 8 7 15 12 15 14 5 10 3 9 0 15 3 13 6 4 12 8 11 3 7 10 6 6 3 4 1 4 13 1 11 6 2 0 13 15 5 12 7 14 8 11 10 5 2 10 5 0 7 4 12 4 5 7 14 12 6 9 9 1 1 8 13 5 11 15 4 11 2 12 12 2 2 13 0 15 6 14 9 8 6 15 7 2 0 11 5 8 12 3 4 9 8 10 0 15 1 3 5 1 14 6 14 13 10 7 7 10 3 4 13 10 0 11 3 4 12 14 8 9 3 14 1 15 1 11 6 9 11 8 10 4 0 13 2 10 2 0 10 1 12 15 11 15 5 2 9 4 9 0 1 7 3 8 7 3 2 10 13 15 12 7 0 6 6 14 12 11 7 4 6 14 13 13 5 5 3 8 5 7 5 3 4 6 15 14 4 12 14 2 12 13 15 8 0 12 1 1 6 4 2 13 10 11 11 9 12 8 4 11 11 10 15 8 2 7 0 2 0 0 10 3 15 3 5 9 6 8 13 10 1 7 1 7 14 14 9 6 3 5 5 9 13 7 14 2 4 11 12 12 5 0 4 15 13 1 7 10 15 2 3 8 2 15 10 10 5 9 6 15 10 13 12 0 1 11 11 6 11 3 14 7 9 4 8 9 2 8 6 0 14 13 7 5 12 3 5 8 6 0 9 4 1 13 1 14 3
WDisc	4.9250	4.7568

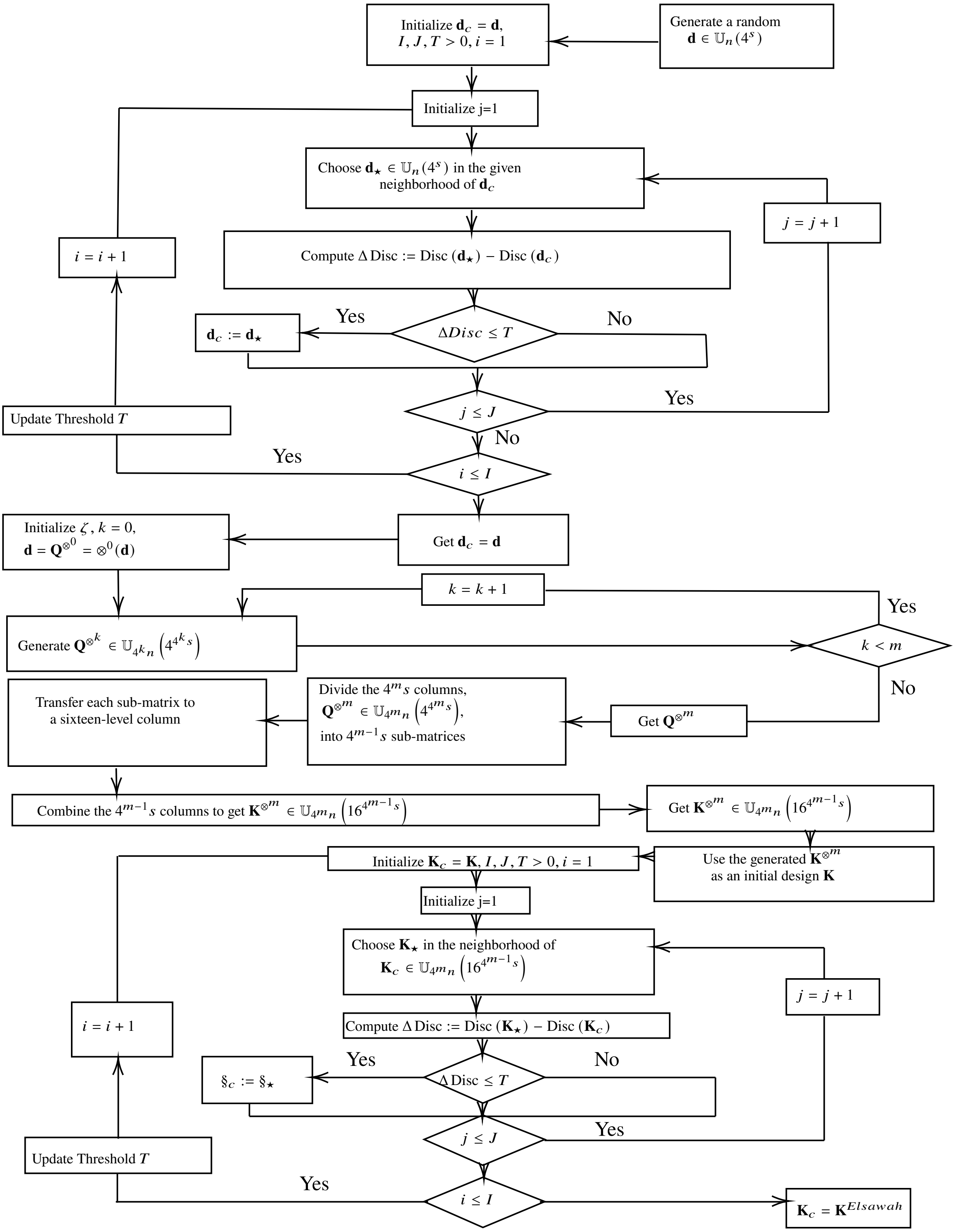


Figure 15: The flowchart of the hybrid algorithm by combining the AMQA [132] with the TA algorithm. Source [132]

Table 40: The designs (datasets) and their corresponding outputs that are used to fit the Wood function. Source [138]

	AMQA[132]					TA using the R-package UniDOE					TA+CGD from [144]					Latin hypercube sample (LHS)				
n	x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4	y
1	0.03125	0.28125	0.53125	0.78125	40.2398	0.0938	0.2812	0.2812	0.7188	55.6098	0.4678	0.6593	0.4717	0.7195	46.0531	0.6184	0.3206	0.3830	0.7843	45.5283
2	0.28125	0.53125	0.78125	0.03125	71.8796	0.5938	0.7188	0.4688	0.9688	65.3334	0.2873	0.3469	0.044	0.7769	70.1574	0.6848	0.1520	0.1785	0.8996	87.6457
3	0.53125	0.78125	0.03125	0.28125	41.9445	0.8438	0.9062	0.7812	0.4062	12.3448	0.1599	0.7847	0.5913	0.5975	67.8505	0.3402	0.6799	0.8624	0.5403	42.1059
4	0.78125	0.03125	0.28125	0.53125	73.1888	0.0312	0.6562	0.5312	0.2188	57.1274	0.0415	0.5965	0.2213	0.3425	55.919	0.3992	0.1893	0.3361	0.2025	27.4699
5	0.09375	0.34375	0.59375	0.84375	40.5487	0.4062	0.1562	0.6562	0.9062	29.6834	0.7808	0.0927	0.3474	0.6514	68.3507	0.2663	0.0725	0.5140	0.1506	33.5069
6	0.34375	0.59375	0.84375	0.09375	74.7163	0.9062	0.2188	0.4062	0.1562	63.0634	0.2165	0.4656	0.4057	0.0389	42.3056	0.1714	0.4336	0.6095	0.9617	52.2108
7	0.59375	0.84375	0.09375	0.34375	41.8396	0.5312	0.3438	0.9688	0.2812	58.3998	0.7252	0.7091	0.9508	0.0981	76.1544	0.7046	0.5451	0.2269	0.8223	58.4052
8	0.84375	0.09375	0.34375	0.59375	76.2757	0.2188	0.0938	0.1562	0.3438	35.1319	0.3397	0.906	0.7747	0.2254	83.2233	0.0179	0.0368	0.4786	0.0339	42.0194
9	0.15625	0.40625	0.65625	0.90625	40.5179	0.2812	0.4688	0.7188	0.0312	59.5152	0.8519	0.53	0.7172	0.8348	17.2176	0.2185	0.7733	0.2839	0.6948	90.5422
10	0.40625	0.65625	0.90625	0.15625	78.4216	0.9688	0.5312	0.5938	0.6562	31.6617	0.6437	0.9514	0.286	0.9514	97.6325	0.9442	0.5675	0.7393	0.6566	17.6727
11	0.65625	0.90625	0 0.15625	0.40625	41.3217	0.7812	0.4062	0.0938	0.8438	73.4283	0.4103	0.0373	0.6566	0.406	26.4865	0.4512	0.9740	0.0783	0.3360	75.1036
12	0.90625	0.15625	0.40625	0.65625	80.4315	0.4688	0.5938	0.0312	0.4688	43.6966	0.5944	0.4045	0.8392	0.5308	14.4977	0.5315	0.2680	0.9938	0.5898	27.5469
13	0.21875	0.46875	0.71875	0.96875	39.9935	0.7188	0.0312	0.8438	0.5312	47.3058	0.9013	0.8316	0.0971	0.4668	24.6248	0.8899	0.8768	0.6318	0.2673	9.7948
14	0.46875	0.71875	0.96875	0.21875	83.1203	0.6562	0.8438	0.2188	0.0938	29.335	0.96	0.2836	0.5329	0.2779	61.6203	0.0833	0.7227	0.0573	0.4658	78.8168
15	0.71875	0.96875	0.21875	0.46875	40.2663	0.1562	0.7812	0.9062	0.7812	60.0546	0.5341	0.2096	0.1563	0.1535	29.7954	0.7513	0.8352	0.8886	0.0759	65.1666
16	0.96875	0.21875	0.46875	0.71875	85.8101	0.3438	0.9688	0.3438	0.5938	95.4985	0.1032	0.1681	0.8969	0.8969	12.8589	0.8676	0.4436	0.7594	0.4096	25.3026

Table 41: The performance of the fitted polynomial models under the above-mentioned criteria for the four designs in Table 40 for the Wood function. Source [138]

	AMQA[132]						TA using the R-package UniDOE					
Measure	F	AIC	BIC	RIC	SCAD	LASSO	F	AIC	BIC	RIC	SCAD	LASSO
MSE	0.2440	0.2000	0.2440	0.2440	15250	0.2560	388.3014	21.3421	21.3421	116.6364	114.2698	135.0445
RMSE	0.4939	0.4472	0.4939	0.4939	123.8951	0.5059	19.7054	4.6197	4.6197	10.7998	10.6897	11.6209
NRMSE	0.8314	0.7528	0.8314	0.8314	208.5522	0.8517	38.5817	9.0451	9.0451	21.1453	20.9297	22.7528
MAE	0.3538	0.4031	0.3538	0.3538	67.1919	0.4244	13.1241	3.3090	3.3090	7.4818	6.8055	9.3676
MRE	0.7342	0.7471	0.7342	0.7342	50.4045	0.7982	31.4153	9.4151	9.4151	22.6772	20.0147	24.5731
R^2	0.9993	0.9994	0.9993	0.9993	-	0.9993	-	0.9450	0.9450	0.6996	0.7057	0.6522
R^2_{adj}	0.9991	0.9992	0.9991	0.9991	-	0.9988	-	0.8626	0.8626	0.5904	0.5586	0.5258
	TA+CGD [144]						Latin hypercube sample (LHS)					
Measure	F	AIC	BIC	RIC	SCAD	LASSO	F	AIC	BIC	RIC	SCAD	LASSO
MSE	479.3988	6.0494	6.0494	8.1765	10.9077	102.4503	364.5458	16.9490	16.9490	40.7637	347.7521	1012.1726
RMSE	21.8952	2.4595	2.4595	2.8594	3.3027	10.1218	19.0931	4.1169	4.1169	6.3846	18.6481	31.8147
NRMSE	44.0798	4.9516	4.9516	5.7567	6.6490	20.3773	39.2237	8.4576	8.4576	13.1162	38.3096	65.3582
MAE	19.5913	1.8637	1.8637	1.9115	2.1309	5.6494	11.1167	2.9587	2.9587	3.5580	11.1491	25.1978
MRE	40.2068	5.6015	5.6015	5.5300	7.4650	16.9071	37.0447	9.7496	9.7496	19.6495	170.2474	2456.2983
R^2	0.2819	0.9909	0.9909	0.9878	0.9837	0.8465	0.3860	0.9715	0.9715	0.9313	0.4143	-
R^2_{adj}	0.2306	0.9773	0.9773	0.9738	0.9650	0.7698	0.3425	0.9465	0.9465	0.8970	0.2679	-

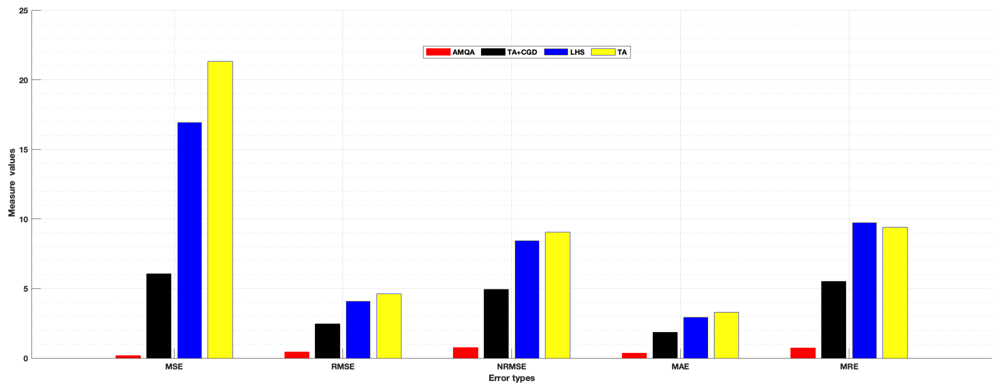


Figure 16: The minimum values for each type of error in Table 2 for the Wood function. Source [138]

6 Conclusion

This selective review has surveyed the most impactful algorithmic and theoretical advances in uniform experimental design over the past decade. Confronted with the growing complexity of modern experiments—including high-dimensional factor spaces, constrained resources, and mixed-level structures—researchers have developed sophisticated tools that significantly enhance the efficiency, flexibility, and theoretical coherence of experimental design. The reviewed progress can be organized around three interconnected pillars: *Enhanced stochastic search algorithms*, such as adjusted threshold accepting and permutation-projection methods, which enable the construction of (nearly) uniform minimum aberration designs. *Fold-over frameworks* for sequential experimentation, which provide a systematic approach to breaking aliasing structures in symmetric and asymmetric designs. *Deterministic construction methods*—including multiple doubling, tripling, quadrupling, and their integrations—that generate large-scale uniform designs for low-level, high-level, and mixed-level factors without exhaustive search.

Collectively, these advances bridge traditional criteria such as uniformity, aberration, and orthogonality, revealing deep theoretical connections and offering a unified basis for design selection. The result is a robust, computationally efficient toolkit that supports experimenters across scientific and industrial domains in designing more informative and cost-effective experiments.

Despite these advances, several challenges and opportunities remain. Future research may focus on: Developing adaptive algorithms for real-time experimental design in streaming or high-throughput settings. Extending fold-over and deterministic frameworks to highly irregular experimental domains or non-standard data types (e.g., functional, graph-based). Integrating machine learning and Bayesian optimization with traditional design criteria to further automate the design selection process. Improving the accessibility and usability of advanced design tools for non-specialist practitioners through software and interactive platforms.

In summary, the past decade has marked a period of remarkable innovation in uniform experimental design. By merging algorithmic ingenuity with theoretical insight, the field has moved beyond conventional methodologies toward a more integrated, efficient, and scalable framework for experimentation. This progress not only enriches the statistical foundations of design but also empowers researchers and engineers to explore increasingly complex systems with greater precision and confidence

Acknowledgments

The authors sincerely thank the referees, Associate Editor, and Editor-in-Chief for their valuable comments and suggestions, which have greatly improved this paper. The authors also acknowledge the use of DeepSeek for assistance in improving the English grammar and language clarity. Furthermore, all figures and tables in this work are reproduced or adapted from the cited references.

Funding

This work was supported by the National Nature Science Foundation of China No. (12371261).

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Huang H., Liu M.Q., Tan M.T., and Fang H.B. (2023). Design and modeling for drug combination experiments with order effects. *Statistics in Medicine* 42(9):1353-1367

[2] Mohamed H.S., Elsawah A.M., Shao Y.B., Wu C.S., and Bakri M. (2023). Analysis on the shear failure of HSS S690-CWGs via mathematical modeling. *Engineering Failure Analysis* 143(A):106881

[3] Iyer A., Yerramilli S., Rondinelli J.M., Apley D.W., and Chen W. (2022). Descriptor ided Bayesian optimization for many-level qualitative variables with materials design applications. *J Mechanical Design* 145

[4] Striegel C., Biehler J., Wall W.A., and Kauermann G. (2022). A multifidelity functionon-function model applied to an abdominal aortic aneurysm. *Technometrics* 112

[5] Prasath B.B., Elsawah A.M., Liyuan Z., and Poon K. (2021). Modeling and optimization of the effect of abiotic stressors on the productivity of the biomass, chlorophyll and lutein in microalgae Chlorella pyrenoidosa. *J Agriculture and Food Research* 5:100163

[6] Zhang W., Wang X., and Po Y.P. (2021). A new design of the continual reassessment method. *Commun Stat Simul Comput* 50(7): 2015-2024

[7] Mak S., Sung C.L., Wang X., Yeh S.T., Chang Y.H., Joseph V.R., Yang V., and Wu C.J. (2018). An efficient surrogate model for emulation and physics extraction of large eddy simulations. *American Statistical Association* 113:1443-1456

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

[9] Joseph V.R. (2016). Space-filling designs for computer experiments: A review. *Quality Engineering*: 28, 28-35

[10] Fang K.T. (1980). The uniform designs: Application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica* 3:363-372

[11] Fang KT., and Zhou Y. (2025). Uniform experimental design. *In: Lovric, M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg*

[12] Fang K.T., Liu M.Q., Qin H., and Zhou Y.D. (2018). Theory and application of uniform experimental designs. *Springer, Singapore*

[13] Fang K.T., and Li R. (2006). Uniform design for computer experiments and its optimal properties. *International J of Materials and Product Technology* 25(1/2/3): 198-210.

[14] Simpson T.W., Lin D.K.J., and Chen W. (2001). Sampling strategies for computer experiments: Design and analysis. *International J of Reliability and Applications* 2(3):209-240

[15] Santner T.J., Williams B.J., and Notz W.I. (2018). The design and analysis of computer experiments (2nd ed.). *Springer*

[16] Elsawah A.M. Fang K.T., and Deng Y.H. (2021). Some interesting behaviors of good lattice point sets. *Commun Stat Simul Comput* 50(11): 3650–3668

[17] Wu C.F.J., and Hamada M. (2000). Experiments: Planning, Analysis and Parameter Design Optimization. *Wiley, New York*

[18] Mukerjee R., and Wu C.F.J. (2006). A Modern Theory of Factorial Designs. *Springer, New York*

[19] Xu H., Phoa F.K.H., and Wong W.K. (2009). Recent developments in non-regular fractional factorial designs. *Statistics Surveys* 3:18-46

[20] Fang K.T., and Hickernell F.J. (1995). The uniform design and its applications. *Bulletin of the International Statistical Institute* 1:333-49

[21] McKay M.D., Beckman R.J., and Conover W.J. (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

[22] Fang K.T., and Mukerjee R. (2000). A connection between uniformity and aberration in regular fractions of two-level factorials. *Biometrika* 87:93-198

[23] Fang K.T., Ma C.X. and Mukerjee R. (2002). Uniformity in fractional factorials. In K. T. Fang, F. J. Hickernell, & H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo methods in scientific computing. *Springer, Berlin.*

[24] Fang K.T., Lu X., and Winker P. (2003). Lower bounds for centered and wrap-around L_2-discrepancies and construction of uniform designs by threshold accepting. *J Complexity* 19:692-711

[25] Qin H., Zhang S.L., and Fang K.T. (2006). Constructing uniform design with two- or three-level. *Acta Math Sci Ser B* 26:451-459

[26] Chatterjee K., Li Z.,and Qin H. (2012). Some new lower bounds to centered and wrap-round L_2-discrepancies. *Statist Probab Lett* 82:1367-1373

[27] Fang K.T., Maringer D., Tang Y., and Winker P. (2005). Lower bounds and stochastic optimization algorithms for uniform designs with three or four levels. *Math Comp* 75:859-878

[28] Fang K.T., Tang Y., and Yin X.J. (2008). Lower bounds of various criteria in experimental designs. *J Statist Plann Inference* 138:184-195

[29] Jones B., Moyer K.A., and Goos P. (2021). A-optimal versus D-optimal design of screening experiments. *J Quality Technology* 53(4):369-381

[30] Ba S., Myers W.R., and Brenneman W.A. (2015). Optimal sliced Latin hypercube designs. *Technometrics* 57(4):479-87

[31] Vazquez A.R., and Xu H. (2024). An integer programming algorithm for constructing maximin distance designs from good lattice point sets. *Stat Sin* 34. Advance online publication. doi: 10.5705/ss.202021.0362.

[32] Xiao Q., and Xu H. (2017). Construction of maximin distance Latin squares and related Latin hypercube designs. *Biometrika* 104(2):455-464

[33] Zhou Y. D., and Xu H. (2015). Space-filling properties of good lattice point sets. *Biometrika* 102(4):959-966

[34] Xu H., and Wu C.F.J. (2001). Generalized minimum aberration for asymmetrical fractional factorial designs. *Ann Statist* 29:549-560

[35] Ma C.X., and Fang K.T. (2001). A note on generalized aberration in factorial designs. *Metrika* 53: 85-93

[36] Tang B., and Deng L.Y. (1999). Minimum G2 -aberration for non-regular fractional factorial designs. *Ann Statist* 27:1914-1926

[37] Xu H., and Wu C.F.J. (2001). Generalized minimum aberration for asymmetrical fractional factorial designs. *Ann Statist* 29:549-560

[38] Hedayat A.S., Sloane N.J., and Stufken J. (1999). Orthogonal Arrays: Theory and Application. *Springer, Berlin*

[39] Bingham D., Sitter R.R., and Tang B. (2009). Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika* 96:51-65

[40] Pang F., Liu M.Q., and Lin D.K.J. (2009). A construction method for orthogonal latin hypercube designs with prime power levels. *Stat Sin* 19:1721-1728

[41] Lu X., Fang K.T., Xu Q.F., and Yin J.X. (2002). Balance pattern and BP-optimal factorial designs. *Technical Report* 324, Hong Kong Baptist University.

[42] Zhang A., Fang K.T., Li R., and Sudjianto A. (2005). Majorization framework for balanced lattice designs. *Ann Stat* 33(6):2837-2853

[43] Liu M.Q., Fang K.T., and Hickernell F.J. (2006). Connection among different criteria for asymmetrical fractional factorial designs. *Statistica Sinica* 16:1285–297

[44] Wang Y., and Fang K.T. (1981). A note on uniform distribution and experimental design. *Chin Sci Bull* 26:485-489.

[45] Fang K.T., Lin D.K.J., Winker P., and Zhang Y. (2000). Uniform design: theory and application. *Technometrics* 42:237-248

[46] Liang Y.Z., Fang K.T., and Xu Q.S. (2001). Uniform design and its applications in chemistry and chemical engineering. *Chemom Intell Lab Syst* 58:43-57

[47] Fang K.T., Li R.Z., Sudjianto A. (2006). Design and modeling for computer experiments. *Chapman and Hall/CRC, New York*

[48] Xu Q.S., Xu Y.D., Li L., and Fang K.T. (2018). Uniform experimental design in chemometrics. *J Chemom* 32:1-10

[49] Weyl, H. (1916). über die gleichverteilung der zahlem mod eins. *Math Ann* 77: 313-352.

[50] Warnock T.T. (1972). Computational investigations of low discrepancy point sets. In Applications of number theory to numerical analysis, ed. S. K. Zaremba, 319–43. *Academic Press, New York*.

[51] Hickernell F.J. (1998). A generalized discrepancy and quadrature error bound. *Math Comp* 67: 299-322

[52] Hickernell F.J. (1998). Lattice rules: how well do they measure up? In: Hellekalek, P., Larcher, G. (Eds.), Random and Quasi-Random Point Sets. In: Lecture notes in Statistics, vol. 138. *Springer, New York* pp. 109-166

[53] Qin H., and Fang K.T. (2004). Discrete discrepancy in factorial designs. *Metrika* 60:59-72.

[54] Zhou Y.D., Ning J.H., and Song X.B. (2008). Lee discrepancy and its applications in experimental designs. *Stat Probab Lett* 78:1933-1942

[55] Zhou Y.D., Fang K.F., and Ning J.H. (2013). Mixture discrepancy for quasi-random point sets. *J Complexity* 29:283-301.

[56] Androulakis E., Drosou K., Koukouvinos C., and Zhou Y.D. (2016). Measures of uniformity in experimental designs: A selective overview. *Commun Stat Theo Meth* 45(13): 3782-3806

[57] Kirkpatrick S., Gelatt C.D., and Vecchi M.P. (1983). Optimization by simulated annealing. *Science* 220(4598):671-680

[58] Winker P., and Fang K.T. (1998). Optimal U-type design, in: Monte Carlo and Quasi-Monte Carlo Methods1996,Vol.436–448 *Springer*

[59] Fang K.T., Ke X., and Elsawah A.M. (2017). Construction of uniform designs via an adjusted threshold accepting algorithm. *J Complexity* 43:28-37

[60] Elsawah A.M., and Qin H. (2014). Lower bound of centered L_2 -discrepancy for four levels U-type designs. *Statist Probab Lett* 93:65-71

[61] Elsawah A.M., and Qin H. (2015). Mixture discrepancy on symmetric balanced designs. *Statist Probab Lett* 104:123-132

[62] Ke X., Zhang R., and Ye H.J. (2015). Two- and three-level lower bounds for mixture L_2 -discrepancy and construction of uniform designs by threshold accepting. *J Complexity* 31(5):741-753

[63] Elsawah A.M., and Qin H. (2015). Lower bound of centered L_2 -discrepancy for mixed two and three levels U-type designs. *J Statist Plann Inference* 161:1–11

[64] Elsawah A.M., and Qin H. (2016). Asymmetric uniform designs based on mixture discrepancy. *J Applied Statistics* 43 (12):2280-2294

[65] Elsawah A.M., Fang K.T., He P. and Qin H. (2021). Sharp lower bounds of various uniformity criteria for constructing uniform designs. *Stat Pap* 62(1):461-1482

[66] Elsawah A.M. (2019). Constructing optimal router bit life sequential experimental designs: new results with a case study. *Commun Stat Simul Comput* 48 (3):723-752

[67] Elsawah A.M. (2018). Choice of optimal second stage designs in two-stage experiments. *Computational Statistics* 33 (2):933–965

[68] Elsawah A.M. (2019). Designing uniform computer sequential experiments with mixture levels using Lee discrepancy. *J Syst Sci Complex* 32:681-708

[69] Elsawah A.M., and Fang K.T. (2020). New foundations for designing U-optimal follow-up experiments with flexible levels. *Stat Pap* 61(2):823–849

[70] Tang Y., Xu H., and Lin D.K.J. (2012). Uniform fractional factorial designs. *Ann Stat* 40:891-907

[71] Tang Y., and Xu H. (2013). An effective construction method for multi-level uniform designs. *J Stat Plan Inference* 143:1583-1589

[72] Xu G., Zhang J., and Tang Y. (2014). Level permutation method for constructing uniform designs under the wrap-around L_2-discrepancy. *J Complex* 30:46-53

[73] Tang Y., and Xu H. (2014). Permuting regular fractional factorial designs for screening quantitative factors. *Biometrika* 101(2):333-350

[74] Tang Y., and Xu H. (2014). Permuting regular fractional factorial designs for screening quantitative factors. *Biometrika* 101(2):333-350

[75] Chen W., Qi Z.F., and Zhou Y.D. (2015). Constructing uniform designs under mixture discrepancy. *Stat Probab Lett* 97:76-82

[76] Elsawah A.M., Fang K.T., and Ke X. (2021). New recommended designs for screening either qualitative or quantitative factors. *Stat Pap* 62(1): 267-307

[77] Lam C., and Tonchev V.D. (1996). Classification of affine resolvable $2-(27,9,4)$ designs. *J Statist Plan Infer* 56:187-202

[78] Chen J., Sun D.X. and Wu C.F.J. (1993). A catalogue of two-level and three-level fractional factorial designs with small runs. *Int Statist Rev* 61:131-135

[79] Evangelaras H., Koukouvinos C., Dean A.M., and Dingus C.A. (2005). Projection properties of certain three level orthogonal arrays. *Metrika* 62:241-257

[80] Xu H. (2005). A catalogue of three-level regular fractional factorial designs. *Metrika* 62:259-281

[81] Zhou Y.D., and Xu H. (2014). Space-filling fractional factorial designs. *J Am Stat Assoc* 109(507):1134-1144

[82] Yang X., Yang G.J., and Su Y.J. (2019). Lower bound of average centered L_2-discrepancy for U-type designs. *Commun Stat Theo Meth* 48(4): 995-1008

[83] Elsawah A.M. (2020). Building some bridges among various experimental designs. *J Korean Stat Soc* 49:55-81

[84] Elsawah A.M., Tang Y., and Fang K.T. (2019). Constructing optimal projection designs. *Statistics* 53(6):1357-1385

[85] Weng L.C., Elsawah A.M., and Fang K.T. (2021). Cross-entropy loss for recommending efficient fold-over technique. *J Syst Sci Complex* 34: 402-439.

[86] Ji Y. B., Alaerts G., Xu C. J.,Hu Y. Z., and Heyden Y. V. (2006). Sequential uniform designs for fingerprints development of Ginkgo biloba extracts by capillary electrophoresis. *J. Chromatography A* 1128:273-281

[87] Tong C. (2006). Refinement strategies for stratified sampling methods. *Reliability Engineering and System Safety* 91:1257-1265

[88] Durrieu G., and L Briollais. (2009). Sequential design for microarray experiments. *J Amer Statist Association* 104:650-660

[89] Loeppky J.L., Moore L.M., and Williams B.J. (2010). Batch sequential designs for computer experiments. *J Statist Plann and Inference* 140:1452-1464

[90] Cheong K.T.W., Htay K., Tan R.H.C., and Lim M. (2012). Identifying combinatorial growth inhibitory effects of various plant extracts on leukemia cells through systematic experimental design. *Amer J Plant Sci* 3:1390-1398

[91] Lu H.M., Ni W.D., Liang W.D., and Man R.L. (2006). Supercritical CO_2 extraction of emodin and physcion from Polygonum cuspidatum and subsequent isolation by semipreparative chromatography. *J Sep Sci* 29:2136–2142

[92] Elsawah A.M., and Qin H. (2015c). A new strategy for optimal foldover two-level designs. *Statist Probab Lett* 103:116-126

[93] Yang F., Zhou Y.D. and Zhang X.R. (2017). Augmented uniform designs. *J Statist Plann Inference* 182:61-73

[94] Box G.E.P., and Hunter J.S. (1961). 2^{k-p} fractional factorial designs. *Technometrics* 3(311-351):449-458

[95] Box G.E.P., Hunter W.G., and Hunter J.S. (1978). Statistics for experimenters. *New York, John Wiley and Sons*

[96] Wu C.F.J. and Hamada M. (2000). Experiments: Planning, Analysis and Parameter Design Optimization. *Wiley, New York*

[97] Montgomery D.C. (2001). Design and analysis of experiments, 5th edn. *Wiley, New York*

[98] Montgomery D.C. and Runger G.C. (1996). Foldovers of 2^{k-p} resolution IV experimental designs. *J Quality Technology* 28(4):446-450

[99] Miller A. and Sitter R.R. (2005). Using folded-over nonorthogonal designs. *Technometrics* 47(4):502-513

[100] Li H., and Mee R.W. (2002). Better foldover fractions for resolution III 2^{k-p} designs. *Technometrics* 44:278-283

[101] Li W. and Lin D.K.J. (2003). Optimal foldover plans for two-level fractional factorial designs. *Technometrics* 45(2):142-149

[102] Li W., Lin D.K.J., and Ye K.Q. (2003). Optimal foldover plans for non-regular orthogonal designs. *Technometrics* 45(4):347-351

[103] Ye K., and Li W. (2003), Some properties for blocked and unblocked foldovers of 2^{k-p} designs. *Stat Sin*13:403-408

[104] Fang K.T., Lin D.K.J. and Qin H. (2003). A note on optimal foldover design. *Statist Probab Lett* 62(3):245-250

[105] Lei Y.J., Qin H., and Zou N. (2010). Some lower bounds of centered L_2-discrepancy on foldover designs. *Acta Math Sci* 30A(6):1555-1561

[106] Ou Z.J., Chatterjee K. and Qin H. (2011). Lower bounds of various discrepancies on combined designs. *Metrika* 74: 109-119

[107] Chatterjee K., Qin H., and Zou N. (2012). Lee discrepancy on two and three mixed level factorials. *Sci China* 55 (3):663-670

[108] Elsawah A.M., and Qin H. (2017). A new look on optimal foldover plans in terms of uniformity criteria. *Commun Stat Theory Meth* 46 (4):1621–1635

[109] Ou Z.J., Qin H., and Cai X. (2015). Optimal foldover plans of three level designs with minimum wrap-around L_2-discrepancy. *Sci China Math* 58:1537-1548

[110] Elsawah A.M. (2017). A closer look at de-aliasing effects using an efficient foldover technique. *Statistics* 51(3):532-557

[111] Elsawah A.M., and Fang K.T. (2019). A catalog of optimal foldover plans for constructing U-uniform minimum aberration four-level combined designs. *J Applied Statist* 46(7): 1288-1322

[112] Phadke M.S. 1986. Design optimization case studies. *AT & T Technical Journal* 65:51-68

[113] Ou Z.J., Qin H., and Cai X. (2014). A lower bound for the wrap-around L_2-discrepancy on combined designs of mixed two- and three-level factorials. *Communi. Statist. Theory Methods* 43:2274-85

[114] Bettonvil, B. , and Kleijnen J.P.C. (1996). Searching for important factors in simulation models with many factors: Sequential bifurcation. *European J Oper Res* 96:180-194

[115] Kleijnen J.P.C., Ham G.V., and Rotmans J. (1992). Techniques for sensitivity analysis of simulation models: A case study of the CO_2 greenhouse effect. *Simulation* 58 (6):410-417

[116] Kleijnen J.P.C., Bettonvil B., and Persson F. (2006). Screening for the important factors in large discrete-event simulation: sequential bifurcation and its applications, in: A. Dean, S. Lewis (Eds.), Screening methods for experimentation in industry, drug discovery, and genetics. *Springer-Verlag., New York* pp. 287–307

[117] Morris M.D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics* 33:161174

[118] Elsawah A.M. (2021). Multiple doubling: a simple effective construction technique for optimal two-level experimental designs. *Stat Pap* 62(6):2923-2967

[119] Plackett R.L., and Burman J.P. (1946). The design of optimum multifactorial experiments. *Biometrika* 33: 305-325

[120] Chen H. and Cheng C.S. (2006). Doubling and projection: A method of constructing two-level designs of resolution IV. *Ann Statist* 34: 546-558

[121] Xu H., and Cheng C.S. (2008). A complementary design theory for doubling. *Ann Stat* 36:445-457

[122] Lei Y.J., and Qin H. (2014). Uniformity in double design. *Acta Math Appl Sin* 30(3):773-780

[123] Zou N., and Qin H. (2017). Some properties of double designs in terms of Lee discrepancy. *Acta Math Sci* 37B(2):477-487

[124] Elsawah A.M. (2021). An appealing technique for designing optimal large experiments with three-level factors. *J Comput Appl Math* 384:113164

[125] Li H., and Qin H. (2018). Some new results on triple designs. *Statist Probab Lett* 139: 1-9 [28]

[126] Elsawah A.M. (2022). Improving the space-filling behavior of multiple triple designs. *Comp Appl Math* 41:180

[127] Elsawah A.M. (2022). Designing optimal large four-level experiments: A new technique without recourse to optimization softwares. *Commun Math Stat* 10:623-652

[128] Li H., and Qin H. (2020). Quadrupling: construction of uniform designs with large run sizes. *Metrika* 83:527-544

[129] Elsawah A.M. (2024). Level permutations and factor projections of multiple quadruple designs. *Commun Stat Simul Comput* 53(10), 4893-4920

[130] Elsawah A.M., and Vishwakarma G.K. (2022). A systematic construction approach for non-regular fractional factorial four-level designs via quaternary linear codes. *Comp Appl Math* 41:323

[131] Elsawah A.M., Wang Y.A., Celem S.M., and Qin H. (2023). A novel technique for constructing nonregular nine-level designs: Adjusted multiple tripling technique. *J Comp Appl Math* 424:115016

[132] Elsawah A.M. (2024). A novel coding scheme for generating sixteen codes from quaternary codes with applications. *Comp Appl Math* 43(3): 118

[133] Elsawah A.M. (2022). A novel non-heuristic search technique for constructing uniform designs with a mixture of two- and four-level factors: a simple industrial applicable approach. *J Korean Stat Soc* 51:716-757

[134] Elsawah A.M. (2023). A novel hybrid algorithm for designing mixed three-and nine-level experiments without modeling assumptions. *Commun Stat Simul Comput* 54(4):1003-1037

[135] Elsawah A.M. (2024). A novel low complexity fast technique for effectively designing mixed-level experiment. *Commun Stat Simul Comput* 53(1): 315-343

[136] Elsawah A.M. (2024). A novel doubling-tripling-threshold accepting hybrid algorithm for constructing asymmetric space-filling designs. *J Korean Stat Soc* 53:1–41

[137] Elsawah A.M. (2022). Novel techniques for performing successful follow-up experiments based on prior information from initial-stage experiments. *Statistics* 56(5):1133-1165

[138] Elsawah A.M., Laala, B., Abdel-Hamid A.H., and Qin H. (2026). Construction of uniform designs for high-dimensional screening via an adjusted multiple quadrupling algorithm. *Commun Stat Simul Comput* 55(1):195-220

[139] Ke X., Fang, K.T., Elsawah A. M., and Lin Y. (2023). New non-isomorphic detection methods for orthogonal designs. *Commun Stat Simul Comput* 52(1):27-42

[140] Weng L.C., Fang K.T., and Elsawah, A.M. (2023). Degree of isomorphism: A novel criterion for identifying and classifying orthogonal designs. *Stat Pap* 64:93–116

[141] Elsawah A.M. and Gong Y. (2023). A new non-iterative deterministic algorithm for constructing asymptotically orthogonal maximin distance Latin hypercube designs. *J Korean Stat Soc* 52(3):621-646

[142] Elsawah A.M. (2026). Constructing orthogonal maximin distance uniform projection designs for computer experiments. *J Comput Appl Math* 473:116902

[143] Zhang A., and Li H. (2017). UniDOE: An R package for constructing uniform design of experiments via stochastic and adaptive threshold accepting algorithm. *Technical Report*

[144] Lai J., Fang K.T., Peng X. and Lin Y. 2024. Construction of uniform designs over continuous domain in computer experiments. *Commun Stat Simul Comput* 53 (1):130-46