An integer programming approach for constructing maximin distance designs from good lattice point sets and the Williams' transformation

Alan R. Vazquez

University of California, Los Angeles

alanrvazquez@stat.ucla.edu

Quality and Productivity Research Conference

June 13-16, 2022

Funded by the Research Foundation – Flanders, Belgium

Outline

1. Introduction
2. Good lattice point sets and the Williams’ transformation
3. Integer programming algorithm
4. Results and discussion

Computer models and space-filling designs

- Computer models allow us to simulate complex physical phenomena.

- Example:
  - Car crash simulations (Oyama et al., 2019)
  - Design optimization of combat drones (Siddiqi & Lee, 2019).

- They involve many parameters (factors) and, often, are computationally expensive!

- To overcome this issue, we conduct a computer experiment to build a computationally-cheap surrogate model.

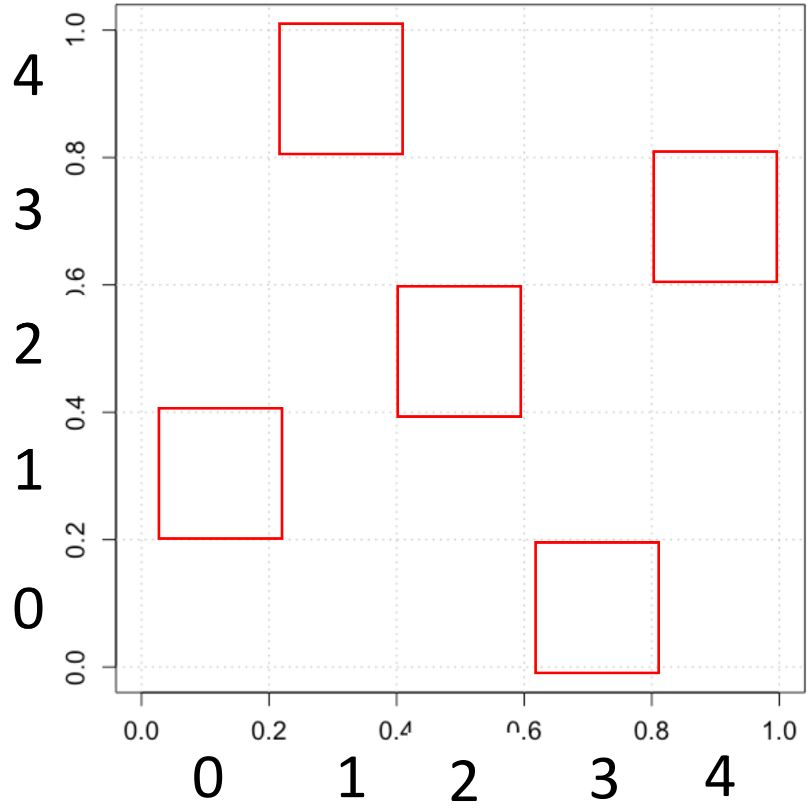
- **Space-filling designs** are attractive for computer experiments because their points (or runs) fill the experimental region uniformly.

Latin hypercube designs (LHDs)

An N-run k-factor LHD is an $N \times k$ matrix whose columns are permutations of the elements in $\{0, \ldots, N-1\}$.

Example: 5-run 2-factor LHD

$$\begin{array}{c} \begin{array}{cc} X_1 & X_2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 4 \\ 2 & 2 \\ 3 & 0 \\ 4 & 3 \end{pmatrix} \end{array}$$

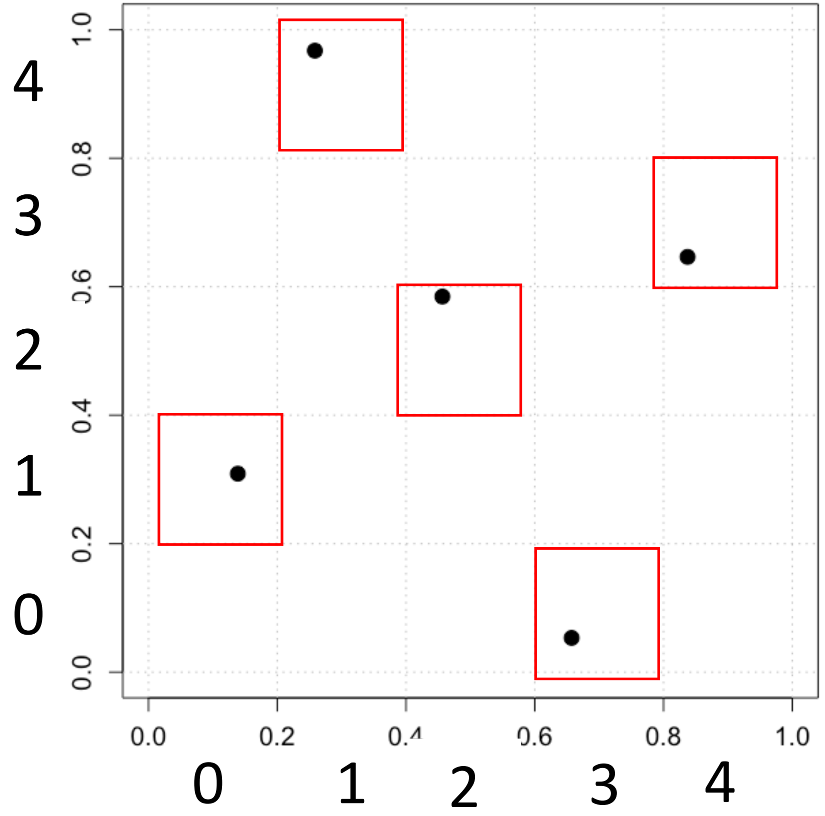


Latin hypercube designs (LHDs)

An N-run k-factor LHD is an $N \times k$ matrix whose columns are permutations of the elements in $\{0, \ldots, N-1\}$.

Example: 5-run 2-factor LHD

$$\begin{array}{cc} X_1 & X_2 \end{array}$$
$$\begin{pmatrix} 0 & 1 \\ 1 & 4 \\ 2 & 2 \\ 3 & 0 \\ 4 & 3 \end{pmatrix}$$



Maximin distance criterion

The maximin distance criterion (Johnson et al., 1990) measures the minimum distance between two rows in an LHD.

Let $\mathbf{x}_i$ be the i-th row of the LHD.

There are two versions:

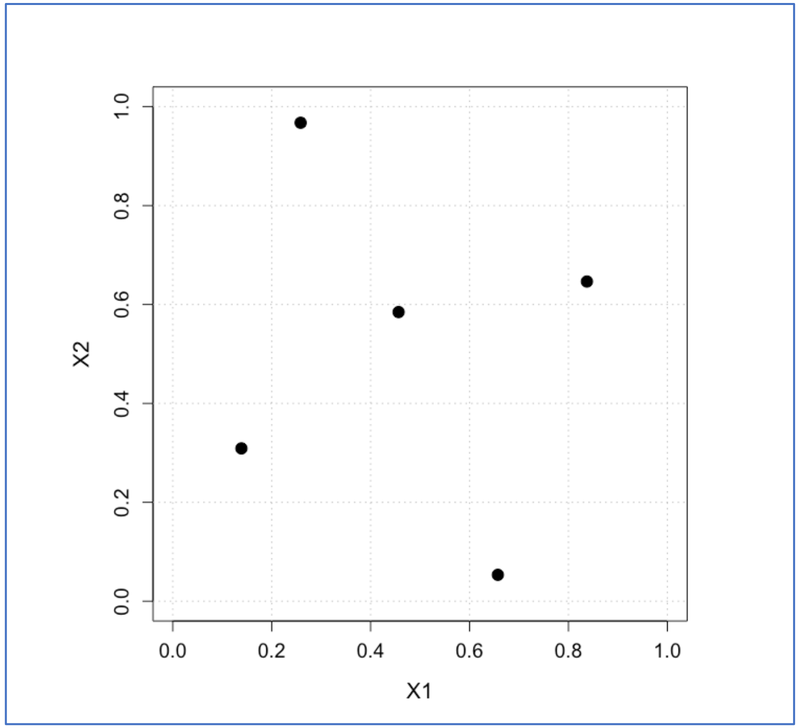
- L_2-distance: $\min\left\{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \;: \text{for all rows } i \text{ and } j\right\}.$
- L_1-distance: $\min\left\{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_1 \;: \text{for all rows } i \text{ and } j\right\}.$

Larger values of the maximin distance criterion are preferred.

Example: Compare two 5-run 2-factor LHDs in terms of the maximin distance criterion.

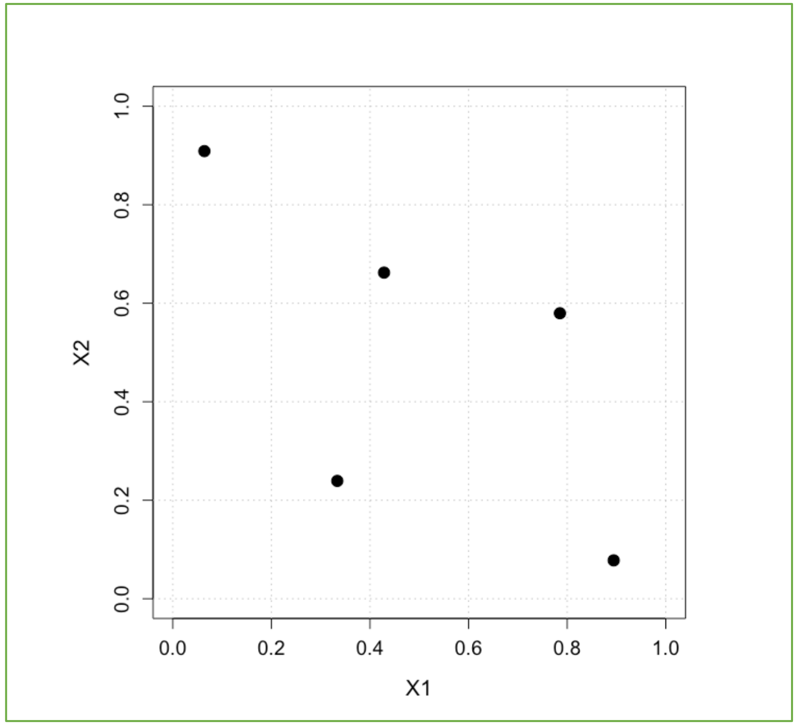
Design 1

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



Design 2

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



Minimum L_1-distance = 3

Minimum L_1-distance = 2

Methods to generate good LHDs

Algorithmic

- Simulated annealing (Morris & Mitchell, 1995; Ba et al., 2015).
- Particle swarm optimization (Chen et al., 2013).
- Iterated local search (Grosso et al., 2009).
- Genetic algorithm (Liefvendahl & Stocki, 2006).

Algebraic

- Good lattice point sets (Zhou & Xu, 2015).
- Williams' transformation (Wang et al., 2018).
- Costas arrays (Xiao & Xu, 2017).
- Orthogonal arrays (Xiao & Xu, 2018).

Among others.

Methods to generate good LHDs

Algorithmic

Limitation:

Computational-performance deteriorates for large number of factors and runs.

Algebraic

Limitation:

Only available for constructing designs of specific sizes.

Outline

1. Introduction
2. Good lattice point sets and the Williams’ transformation
3. Integer programming algorithm
4. Results and discussion

Good lattice point (GLP) set

Let $\phi(N)$ be the number of positive integers smaller than and coprime to *N*.

Definition: A GLP set is an $N \times \phi(N)$ matrix whose columns are permutations of the elements in $\{0, \dots, N-1\}$.

Example: Construct a GLP set $\mathbf{X}$ with *N* = 7 rows and $\phi(N)$ = 6 columns.

Step 1. Write down the positive integers that are smaller and coprime to *N*: $h_1 = 1,\ h_2 = 2, \dots,\ h_{\phi(N)} = 6.$

Step 2. Set the elements of $\mathbf{X}$ as $x_{ij} = ih_j \pmod N$ for $i = 1, \ldots, N$ and $j = 1, \ldots, \phi(N)$.

	h_j values					
i	**1**	**2**	**3**	**4**	**5**	**6**
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1
7	0	0	0	0	0	0

$\mathbf{X}$ is a Latin hypercube design:

- 7 runs and 6 factors
- Minimum L_1-distance = 12

Linear permutations

Zhou and Xu (2015) show that linear permutations of the columns of a GLP set $\mathbf{X}$ may produce a better LHD in terms of the L_1-distance.

Example (cont.): Consider the linear permutation $\mathbf{X} + 4 \pmod 7$.

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 1 & 3 & 5 \\ 3 & 6 & 2 & 5 & 1 & 4 \\ 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 3 & 1 & 6 & 4 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{X} + 4 \pmod 7 = \begin{bmatrix} 5 & 6 & 0 & 1 & 2 & 3 \\ 6 & 1 & 3 & 5 & 0 & 2 \\ 0 & 3 & 6 & 2 & 5 & 1 \\ 1 & 5 & 2 & 6 & 3 & 0 \\ 2 & 0 & 5 & 3 & 1 & 6 \\ 3 & 2 & 1 & 0 & 6 & 5 \\ 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Minimum L_1-distance = 12

Minimum L_1-distance = 13

Williams' transformation

Wang et al. (2018) show that the performance of linearly permuted GLP set can be further improved using the Williams' transformation.

$$W(x) = \begin{cases} 2x & \text{for } 0 \leq x < \frac{N}{2} \\ 2(N-x)-1 & \text{for } \frac{N}{2} \leq x \leq N \end{cases}.$$

The Williams' transformation is a permutation of $\{0, \ldots, N-1\}$.

Example (cont.): Apply the Williams' transformation to each element in $\mathbf{X}+4 \pmod 7$.

$$\mathbf{X}+4 \pmod 7 = \begin{bmatrix} 5 & 6 & 0 & 1 & 2 & 3 \\ 6 & 1 & 3 & 5 & 0 & 2 \\ 0 & 3 & 6 & 2 & 5 & 1 \\ 1 & 5 & 2 & 6 & 3 & 0 \\ 2 & 0 & 5 & 3 & 1 & 6 \\ 3 & 2 & 1 & 0 & 6 & 5 \\ 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Minimum L_1-distance = 13

$$W(\mathbf{X}+4 \pmod 7) = \begin{bmatrix} 3 & 1 & 0 & 2 & 4 & 6 \\ 1 & 2 & 6 & 3 & 0 & 4 \\ 0 & 6 & 1 & 4 & 3 & 2 \\ 2 & 3 & 4 & 1 & 6 & 0 \\ 4 & 0 & 3 & 6 & 2 & 1 \\ 6 & 4 & 2 & 0 & 1 & 3 \\ 5 & 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

Minimum L_1-distance = 16

A general construction method

Goal: Construct a good N-run LHD with $\phi(N)$ factors in terms of maximin distance criterion.

GLP set | Linear permutations | Williams' transformation

$$\mathbf{X} \to \mathbf{X} + 0 \pmod{N} \to \mathbf{Z}_0 = W(\mathbf{X}+0) \pmod{N}$$

$$\mathbf{X} \to \mathbf{X} + 1 \pmod{N} \to \boxed{\mathbf{Z}_1} = W(\mathbf{X}+1) \pmod{N}$$ **Select best**

$$\vdots \qquad \vdots$$

$$\mathbf{X} \to \mathbf{X} + N - 1 \pmod{N} \to \mathbf{Z}_{N-1} = W(\mathbf{X}+N-1) \pmod{N}$$

Wang et al. (2018)

Research question

+ GLP sets, linear permutations and the Williams' transformation can generate attractive LHDs (Wang et al., 2018).

- However, the method is limited to LHDs with $\phi(N)$ factors. For N = 30, we can only construct 8-factor LHDs.
- It is unknown how to generate LHDs with more or fewer factors than $\phi(N)$ using the method.

In this talk, we introduce an integer programming algorithm for generating LHDs with flexible numbers of factors.

Outline

1. Introduction
2. Good lattice point sets and the Williams' transformation
3. Integer programming algorithm
4. Results and discussion

Integer programming (IP) algorithm

Integer programming is an optimization method to determine the values of a set of discrete decision variables so as to maximize or minimize an objective function, while satisfying a set of linear constraints (Wolsey, 2020).

Our IP algorithm consists of

- A candidate set of attractive columns.
- A problem formulation for finding maximin distance LHDs.
- The use of state-of-the-art optimization software to solve this problem formulation.

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs.

Step 1. Construct the five $N \times \phi(N)$ designs using the GLP set, linear permutations and the Williams's transformation.

$$\mathbf{Z}_0 = \begin{pmatrix} 2 & 4 & 3 & 1 \\ 4 & 1 & 2 & 3 \\ 3 & 2 & 1 & 4 \\ 1 & 3 & 4 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{Z}_1 = \begin{pmatrix} 4 & 3 & 1 & 0 \\ 3 & 0 & 4 & 1 \\ 1 & 4 & 0 & 3 \\ 0 & 1 & 3 & 4 \\ 2 & 2 & 2 & 2 \end{pmatrix} \qquad \mathbf{Z}_2 = \begin{pmatrix} 3 & 1 & 0 & 2 \\ 1 & 2 & 3 & 0 \\ 0 & 3 & 2 & 1 \\ 2 & 0 & 1 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$\mathbf{Z}_3 = \begin{pmatrix} 1 & 0 & 2 & 4 \\ 0 & 4 & 1 & 2 \\ 2 & 1 & 4 & 0 \\ 4 & 2 & 0 & 1 \\ 3 & 3 & 3 & 3 \end{pmatrix} \qquad \mathbf{Z}_4 = \begin{pmatrix} 0 & 2 & 4 & 3 \\ 2 & 3 & 0 & 4 \\ 4 & 0 & 3 & 2 \\ 3 & 4 & 2 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs.

Step 2. Construct an initial candidate set by concatenating the designs.

$$\mathbf{C} = \begin{array}{c} \begin{array}{ccccc} \quad\mathbf{Z}_0\quad & \quad\mathbf{Z}_1\quad & \quad\mathbf{Z}_2\quad & \quad\mathbf{Z}_3\quad & \quad\mathbf{Z}_4\quad \end{array} \\ \left(\begin{array}{cccc|cccc|cccc|cccc|cccc} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 3 & 1 & 0 & 2 & 1 & 0 & 2 & 4 & 0 & 2 & 4 & 3 \\ 4 & 1 & 2 & 3 & 3 & 0 & 4 & 1 & 1 & 2 & 3 & 0 & 0 & 4 & 1 & 2 & 2 & 3 & 0 & 4 \\ 3 & 2 & 1 & 4 & 1 & 4 & 0 & 3 & 0 & 3 & 2 & 1 & 2 & 1 & 4 & 0 & 4 & 0 & 3 & 2 \\ 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \end{array}\right) \end{array}$$

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with $N = 5$ runs.

Step 2. Construct an initial candidate set by concatenating the designs.

$$\mathbf{C} = \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 3 & 1 & 0 & 2 & 1 & 0 & 2 & 4 & 0 & 2 & 4 & 3 \\ 4 & 1 & 2 & 3 & 3 & 0 & 4 & 1 & 1 & 2 & 3 & 0 & 0 & 4 & 1 & 2 & 2 & 3 & 0 & 4 \\ 3 & 2 & 1 & 4 & 1 & 4 & 0 & 3 & 0 & 3 & 2 & 1 & 2 & 1 & 4 & 0 & 4 & 0 & 3 & 2 \\ 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Fully correlated columns: $\begin{pmatrix} 2 \\ 4 \\ 3 \\ 1 \\ 0 \end{pmatrix} = 4 - \begin{pmatrix} 2 \\ 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}$ (mod 5)

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs.

Step 2. Construct an initial candidate set by concatenating the designs.

$$\mathbf{C} = \begin{pmatrix} 2 & \boxed{4} & 3 & 1 & 4 & 3 & 1 & 0 & 3 & 1 & \boxed{0} & 2 & 1 & 0 & 2 & 4 & 0 & 2 & 4 & 3 \\ 4 & \boxed{1} & 2 & 3 & 3 & 0 & 4 & 1 & 1 & 2 & \boxed{3} & 0 & 0 & 4 & 1 & 2 & 2 & 3 & 0 & 4 \\ 3 & \boxed{2} & 1 & 4 & 1 & 4 & 0 & 3 & 0 & 3 & \boxed{2} & 1 & 2 & 1 & 4 & 0 & 4 & 0 & 3 & 2 \\ 1 & \boxed{3} & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & \boxed{1} & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 \\ 0 & \boxed{0} & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & \boxed{4} & 4 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Fully correlated columns: $\begin{pmatrix} 4 \\ 1 \\ 2 \\ 3 \\ 0 \end{pmatrix} = 4 - \begin{pmatrix} 0 \\ 3 \\ 2 \\ 1 \\ 4 \end{pmatrix}$ (mod 5)

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs

Step 2. Construct an initial candidate set by concatenating the designs.

$$\mathbf{C} = \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 3 & 1 & 0 & 2 & 1 & 0 & 2 & 4 & 0 & 2 & 4 & 3 \\ 4 & 1 & 2 & 3 & 3 & 0 & 4 & 1 & 1 & 2 & 3 & 0 & 0 & 4 & 1 & 2 & 2 & 3 & 0 & 4 \\ 3 & 2 & 1 & 4 & 1 & 4 & 0 & 3 & 0 & 3 & 2 & 1 & 2 & 1 & 4 & 0 & 4 & 0 & 3 & 2 \\ 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 & 1 & 3 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Fully correlated columns: $\begin{pmatrix} 3 \\ 2 \\ 1 \\ 4 \\ 0 \end{pmatrix} = 4 - \begin{pmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 4 \end{pmatrix}$ (mod 5)

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs.

Step 3. Remove fully correlated columns.

$$\mathbf{C} = \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 2 & 4 \\ 4 & 1 & 2 & 3 & 3 & 0 & 0 & 4 & 1 & 2 \\ 3 & 2 & 1 & 4 & 1 & 4 & 2 & 1 & 4 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 & 4 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}$$

Candidate set: Construction by example

Example: Construct a candidate set for LHDs with N = 5 runs.

Final candidate set with $\phi(N)N/2$ columns

$$\mathbf{C} = \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 2 & 4 \\ 4 & 1 & 2 & 3 & 3 & 0 & 0 & 4 & 1 & 2 \\ 3 & 2 & 1 & 4 & 1 & 4 & 2 & 1 & 4 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 & 4 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}$$

Theorem. If N is an odd prime, then $\mathbf{C}$ is a maximin L_1-distance LHD.

Problem formulation: Encoding of LHDs

$$\mathbf{C} = \begin{array}{c} \begin{matrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} \end{matrix} \\ \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 2 & 4 \\ 4 & 1 & 2 & 3 & 3 & 0 & 0 & 4 & 1 & 2 \\ 3 & 2 & 1 & 4 & 1 & 4 & 2 & 1 & 4 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 & 4 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix} \end{array}$$

The variables y_u are binary:

- $y_u = 1$ if the column is included in the LHD.
- $y_u = 0$ otherwise.

Let k be the number factors in the LHD. We have that

$$\sum_{u=1}^{\phi(N)N/2} y_u = k \, .$$

Calculation of minimum distance

Let c_{iu} denote the element in the *i*-th row and *u*-th column $\mathbf{C}$.

- The L_1-distance between the *i*-th and *j*-th rows in candidate set is

$$\sum_{u=1}^{\phi(N)N/2} \left| c_{iu} - c_{ju} \right|.$$

$$\mathbf{C} = \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 2 & 4 \\ 4 & 1 & 2 & 3 & 3 & 0 & 0 & 4 & 1 & 2 \\ 3 & 2 & 1 & 4 & 1 & 4 & 2 & 1 & 4 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 & 4 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}$$

- The L_1-distance between *i*-th and *j*-th rows in the LHD is

$$\sum_{u=1}^{\phi(N)N/2} \left| c_{iu} - c_{ju} \right| y_u .$$

$$\begin{array}{cc} & \begin{array}{cccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \\ \mathbf{C} = & \begin{pmatrix} 2 & 4 & 3 & 1 & 4 & 3 & 1 & 0 & 2 & 4 \\ 4 & 1 & 2 & 3 & 3 & 0 & 0 & 4 & 1 & 2 \\ 3 & 2 & 1 & 4 & 1 & 4 & 2 & 1 & 4 & 0 \\ 1 & 3 & 4 & 2 & 0 & 1 & 4 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix} \end{array}$$

The final problem formulation

$$\max_{y_u, t} \quad t$$

Subject to:

(1). $\sum_{u=1}^{\phi(N)N/2} |c_{iu} - c_{ju}| y_u \geq t$
for *all* pairs of rows i and j

(2). $\sum_{u=1}^{\phi(N)N/2} y_u = k$

(3). $t \in \mathbb{N}$

(4). $y_u \in \{0, 1\}$

Solved by optimization solvers:
Gurobi, CPLEX or SCIP.

<u>Attractive features</u>:

- Find high-quality designs.
- Provide certificates of optimality.

Outline

1. Introduction
2. Good lattice point sets and the Williams’ transformation
3. Integer programming algorithm
4. Results and discussion

Numerical comparisons

We obtained design problems with 7 to 30 runs and 4 to 28 factors from Wang et al. (2018).

Construction methods:

- IP algorithm with Gurobi v9 and a maximum search time of 5 min.
- SA: Simulated annealing algorithm with 100 iterations (Ba et al., 2015).
- GA: Genetic algorithm with 100 generations (Liefvendahl and Stocki, 2006)
- WXX: GLP, linear permutations and William’s transformation (Wang et al., 2018).
- XX: Costas arrays (Xiao and Xu, 2017).

Results I

		Minimum L_1-distance				
Runs	Factors	IP	SA	GA	WXX	XX
7	6	**16**	15	15	**16**	14
8	4	**11**	**11**	10	10	
9	6	17	**18**	17	16	
10	4	11	11	**12**	11	
11	10	**39**	36	38	**39**	34
12	4	**13**	**13**	**13**	10	
13	12	**54**	52	52	52	48
14	6	**24**	23	**24**	**24**	
15	8	36	35	**37**	36	
16	8	**43**	37	39	36	
17	16	**94**	86	89	**94**	86
18	6	28	28	**30**	28	

Larger the better

Results I

		Minimum L_1-distance				
Runs	Factors	IP	SA	GA	WXX	XX
7	6	**16**	15	15	**16**	14
8	4	**11**	**11**	10	10	
9	6	17	**18**	17	16	
10	4	11	11	**12**	11	
11	10	**39**	36	38	**39**	34
12	4	**13**	**13**	**13**	10	
13	12	**54**	52	52	52	48
14	6	**24**	23	**24**	**24**	
15	8	36	35	**37**	36	
16	8	**43**	37	39	36	
17	16	**94**	86	89	**94**	86
18	6	28	28	**30**	28	

Larger the better

Results II

		Minimum L_1-distance				
Runs	Factors	IP	SA	GA	WXX	XX
19	18	**118**	108	110	115	106
20	8	**47**	43	46	42	
21	12	**77**	73	**77**	76	
22	10	**68**	61	64	**68**	
23	22	**172**	160	161	168	158
24	8	53	50	**54**	36	
25	20	**163**	153	153	162	
26	12	**98**	87	91	**98**	
27	18	**157**	145	147	156	
28	12	**104**	92	97	94	
29	28	270	254	254	**274**	250
30	8	**63**	57	**63**	61	

Larger the better

Results II

		Minimum L_1-distance				
Runs	Factors	IP	SA	GA	WXX	XX
19	18	**118**	108	110	115	106
20	8	**47**	43	46	42	
21	12	**77**	73	**77**	76	
22	10	**68**	61	64	**68**	
23	22	**172**	160	161	168	158
24	8	53	50	**54**	36	
25	20	**163**	153	153	162	
26	12	**98**	87	91	**98**	
27	18	**157**	145	147	156	
28	12	**104**	92	97	94	
29	28	270	254	254	**274**	250
30	8	**63**	57	**63**	61	

Larger the better

Discussion

- Our IP algorithm constructs LHDs that are at least as good as the benchmark methods for 75% of the design problems.
- For larger-sized problems, we propose two modifications to the IP algorithm which allow us to construct LHDs with up to 72 factors and up to 113 runs.
- Use integer programming to construct LHDs that optimize other statistical criteria such as the MaxPro criterion (Joseph et al, 2015).

Vazquez, A. R. and Xu, H. (2022). An integer programming approach for constructing maximin distance designs from good lattice point sets and the Williams' transformation. Working paper.