

Space-Filling Designs

C. F. Jeff Wu

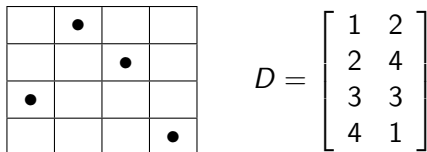
Georgia Institute of Technology

Most commonly used designs for computer experiments

- 1 Latin Hypercube Design
 - Latin Hypercube Design (LHD)
 - Orthogonal Array-based Latin Hypercube Design (OA-LHD)
- 2 Designs Based on Measures of Distance
 - miniMax Distance Design (mM Design)
 - Maximin Distance Design (Mm Design)
- 3 Maximin Latin Hypercube Design (MmLHD)
- 4 Maximum Projection Design (MaxPro)

Latin Hypercube Design (LHD) - Introduction

- Mckay, Beckman and Conover (1979).
- An example: $p = 2, n = 5$:



- Each row and column has one and only one point.
- Each factor has n levels.

Factorial Design 2^2

- On the other hand, a factorial 2^2 design is constructed as

•			•
•			•

$$D = \begin{bmatrix} 1 & 1 \\ 1 & 4 \\ 4 & 1 \\ 4 & 4 \end{bmatrix}$$

- If x_1 (or x_2) is not significant, replication is wasted for computer experiments.
- “Effect sparsity” principle: only a few factors are expected to be important.

Latin Hypercube Design (LHD) - Definition

Definition

A Latin hypercube design (LHD) with n runs and p inputs variables, denoted by $\text{LHD}(n, p)$, is an $n \times p$ matrix, in which each column is a random permutation of $\{1, 2, \dots, n\}$.

Construct an LHD(n, p)

- How to construct an LHD(n, p)?

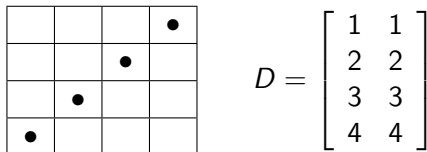
Step1 Randomly permute $\{1, 2, \dots, n\}$ for each x_1, \dots, x_p .

$$D = \begin{bmatrix} x_1 & x_2 & x_3 \\ 1 & 2 & 7 \\ 2 & n & 9 \\ 3 & 5 & 1 \\ \vdots & \vdots & \vdots \\ n & 3 & 2 \end{bmatrix}$$

Step2 $D' \leftarrow \frac{D-0.5}{n}$, where $D \in \{1, 2, \dots, n\}^p$. Thus, $D' \in [0, 1]^p$.

Optimal LHD(n, p)

- Thus, there are $(n!)^{p-1}$ LHDs.
- Not all of them are good. For example,



- This design is **perfectly correlated** and **not space-filling**.
- Finding an “optimal” LHD is a challenge.

Orthogonal Array-based LHD (OA-LHD) - Introduction

- Owen (1992) and Tang (1993).
- **LHD**: one-dimensional balancing property but two and higher dimensional projections can be **very bad**.
- Orthogonal Array (OA) of strength 2 has two dimensional balancing property, so use it to generate an LHD.

Orthogonal Array-based LHD (OA-LHD) - Example

- For example, $n = 9, p = 2$,

OA

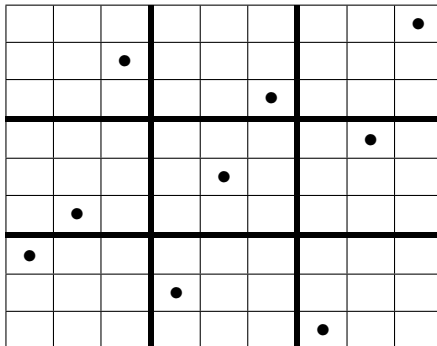
x_1	x_2
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

$1 \rightarrow \{1, 2, 3\} \rightarrow \{3, 2, 1\}$
 $2 \rightarrow \{4, 5, 6\} \rightarrow \{4, 5, 6\}$
 $3 \rightarrow \{7, 8, 9\} \rightarrow \{8, 7, 9\}$

OA-LHD

x_1	x_2
1	3
2	4
3	8
4	2
5	5
6	7
7	1
8	6
9	9

Orthogonal Array-based LHD (OA-LHD) - Example



Latin Hypercube Design (LHD) - Properties

- McKay, Beckman, and Conover (1979) introduced Latin hypercube sampling and subsequently other authors (for example, Stein (1987) and Owen (1992)) have explored their properties. Roughly speaking, suppose we want to find the mean of some known function $G(y(x))$ over X . Then the sample mean of $G(y(x))$ computed from a Latin hypercube design usually has smaller variance than the sample mean computed from a simple random sample.

Designs Based on Measures of Distance

- Johnson, Moore, and Ylvisaker (1991).
- Let $\rho(\cdot, \cdot)$ be a metric.

$$\rho(x_1, x_2) = \rho(x_2, x_1)$$

$$\rho(x_1, x_2) \geq 0$$

$$\rho(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$$

$$\rho(x_1, x_2) \leq \rho(x_1, x_3) + \rho(x_3, x_2)$$

- For example,

$$\rho(u, v) = \left\{ \sum_{j=1}^p |u_j - v_j|^k \right\}^{1/k}$$

- $k = 1$: rectangular distance
- $k = 2$: Euclidean distance

miniMax Distance Design (mM Design) - Introduction

- Let $\rho(x, D) = \min_{x_i \in D} \rho(x, x_i)$ be the minimum distance to the design.
- Let $\chi = [0, 1]^p$ and

$$h = \max_{x \in \chi} \rho(x, D)$$

be the maximum distance in χ .

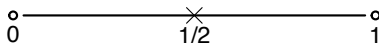
- h is called *fill distance*.
 - the largest gap
 - the radius of the largest ball that can be placed in χ which does not contain any point in D .
- Thus, find a D to minimize h . That is,

$$\min_D \max_{x \in \chi} \rho(x, D)$$

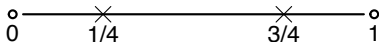
→ miniMax distance design (mM).

miniMax Distance Design (mM Design) - Examples

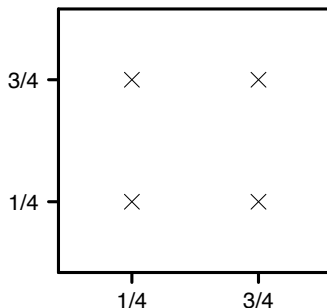
- $p = 1, n = 1$:



- $p = 1, n = 2$:



- $p = 2, n = 4$:



miniMax Distance Design (mM Design) - Properties

- mM designs ensure that all points in χ are not too far from the design.
- Consider the owner of a petroleum corporation who wants to open some gas stations. mM design ensures that no customer is too far from one of the company's gas stations.

Maximin Distance Design (Mm Design) - Introduction

- The minimum distance between any two points in D is

$$2q = \min_{x_1, x_2 \in D} \rho(x_1, x_2),$$

where q is the separation distance or packing radius - the radius of the largest ball that can be placed around every design point such that no two balls overlap.

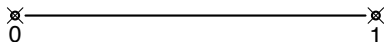
- A large q ensures numerical stability in Kriging.
- A large q tends to decrease h .
- Thus, find a D to maximize $2q$. That is,

$$\max_D \min_{x_1, x_2 \in D} \rho(x_1, x_2)$$

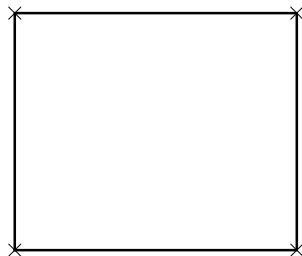
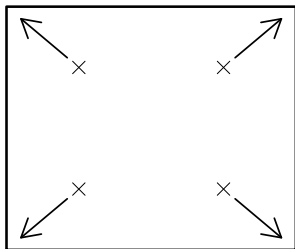
→ Maximin distance design (Mm).

Maximin Distance Design (Mm Design) - Example

- $p = 1, n = 2$:



- $p = 2, n = 4$:

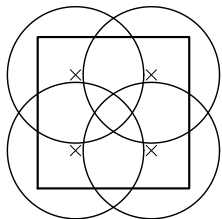


Maximin Distance Design (Mm Design) - Properties

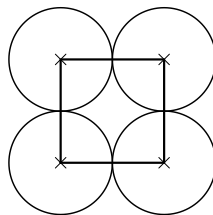
- Mm designs ensure that the points in D are as far apart from each other as possible.
- Gas station example: Mm design ensures that no two gas stations are too close to each other. It minimizes the competition from each other by locating the stations as far apart as possible.

Summary

- In mathematics,
 - covering problems: cover \mathbb{R}^p with spheres. \rightarrow miniMax
 - packing problems: pack spheres in \mathbb{R}^p . \rightarrow Maximin
- Experimental design problem is different because we need to cover $[0, 1]^p$ or pack in $[0, 1]^p$. \rightarrow This introduces boundary effects.



(a) Minimum radius.



(b) Maximum radius.

Summary

- Computationally,

- miniMax:

$$\min_D \max_{\mathbf{x} \in \mathcal{X}} \min_{x_i \in D} \rho(\mathbf{x}, x_i) \rightarrow \text{very hard}$$

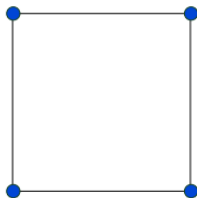
- Maximin:

$$\max_D \min_{\mathbf{x}_1, \mathbf{x}_2 \in D} \rho(\mathbf{x}_1, \mathbf{x}_2) \rightarrow \text{hard, but easier than miniMax}$$

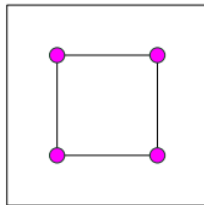
- Mak and Joseph (2017): a new hybrid algorithm combining particle swarm optimization and clustering for generating minimax designs on any convex and bounded design space.
- R package: `minimaxdesign`

Maximin Latin Hypercube Design (MmLHD) - Introduction

- Morris and Mitchell (1995).
- Advantage of mM/Mm designs: run size flexibility and “optimal”.
- Disadvantage of mM/Mm designs: projections are **poor**.



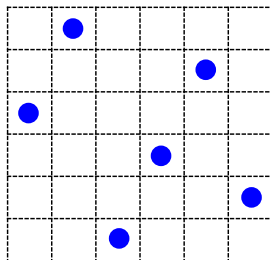
(a) Maximin design



(b) Minimax design.

Maximin Latin Hypercube Design (MmLHD) - Introduction

- LHD: good 1-dimensional projections, but can be poor in terms of space-filling in higher dimensions.



Maximin Latin Hypercube Design (MmLHD) - Introduction

- Combine these two ideas \rightarrow Maximin Latin hypercube designs (MmLHD)

$$\max_D \min_{x_1, x_2 \in D} \rho(x_1, x_2),$$

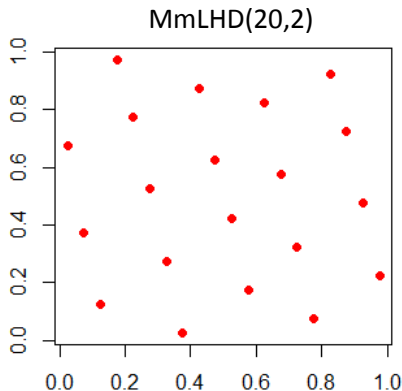
where D is an LHD.

- It is the same as

$$\min_D \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{\rho^k(x_i, x_j)} \right)^{1/k}$$

and $k \rightarrow \infty$, where D is an LHD.

Maximin Latin Hypercube Design (MmLHD) - Examples



Maximum Projection Design (MaxPro) - Introduction

- Joseph, Gul, and Ba (2015).
- MmLHDs only ensure good space-fillingness in p dimensions and uniform projections in a single dimension, but projection properties in $2, 3, \dots, p - 1$ dimensions may not be good.
- In practice, we often have $1 < \text{the number of important factors} < p$.

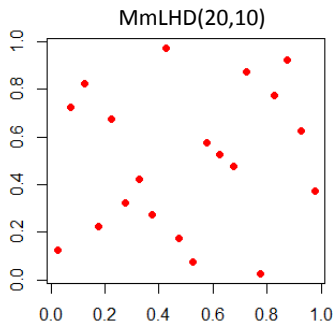


Figure : Projection in two dimensions for a 10 dimensional MmLHD.

Maximum Projection Design (MaxPro)

- How to calculate distances in a projected subspace? **Answer:** Put weights of 1 on the defining factors and 0 on the remaining factors.
- **Weighted Euclidean Distance:**

$$d(x_i, x_j; \boldsymbol{\theta}) = \left\{ \sum_{l=1}^p \theta_l (x_{il} - x_{jl})^2 \right\}^{1/2}.$$

Let $0 \leq \theta_l \leq 1$ be the weight assigned to the factor l and let $\sum_{l=1}^p \theta_l = 1$.

- Then the MmLHD criterion can be modified to

$$\min_D \phi_k(D; \boldsymbol{\theta}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{d^k(x_i, x_j; \boldsymbol{\theta})},$$

where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{p-1})'$ and $\theta_p = 1 - \sum_{l=1}^{p-1} \theta_l$.

Maximum Projection Design (MaxPro)

- In practice, we often have no idea about the importance of the factors before the experiment.
- Assigning non-informative prior:

$$p(\boldsymbol{\theta}) = \frac{1}{(p-1)!}, \boldsymbol{\theta} \in S_{p-1},$$

where $S_{p-1} = \{\boldsymbol{\theta} : \theta_1, \dots, \theta_{p-1} \geq 0, \sum_{i=1}^{p-1} \theta_i \leq 1\}$.

- The design criterion becomes

$$\min_D \mathbb{E}\{\phi_k(D; \boldsymbol{\theta})\} = \int_{S_{p-1}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{d^k(x_i, x_j; \boldsymbol{\theta})} p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

Maximum Projection Design (MaxPro) - New Criterion

Theorem

If $k = 2p$, then under the given prior

$$\mathbb{E}\{\phi_k(D; \theta)\} = \frac{1}{\{(p-1)!\}^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{\prod_{l=1}^p (x_{il} - x_{jl})^2}.$$

- Therefore, we propose a new criterion

$$\min_D \psi(D) = \left(\frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{\prod_{l=1}^p (x_{il} - x_{jl})^2} \right)^{1/p}.$$

- It is called *MaxPro Criterion*.

Maximum Projection Design (MaxPro)

- **MaxPro Criterion:**

$$\min_D \psi(D) = \left(\frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{\prod_{l=1}^p (x_{il} - x_{jl})^2} \right)^{1/p}.$$

- **Maximum projection (MaxPro) design:** maximizes space-filling properties on projections to all possible subsets of factors.
- MaxPro criterion can be computed at a cost no more than a design criterion that ignores projection properties.
- For any l , if $x_{il} = x_{jl}$, then $\psi(D) = \infty$; $\rightarrow n$ distinct levels for each factor \rightarrow **MaxPro designs automatically have the LHD property!**

Maximum Projection Design (MaxPro) - Algorithm

- Design construction algorithm:

Step1 Find an optimal MaxProLHD with simulated annealing (Morris and Mitchell 1995).

Step2 Apply a fast derivative-based continuous optimization algorithm to find locally optimal MaxPro design in the neighborhood of the optimal MaxProLHD:

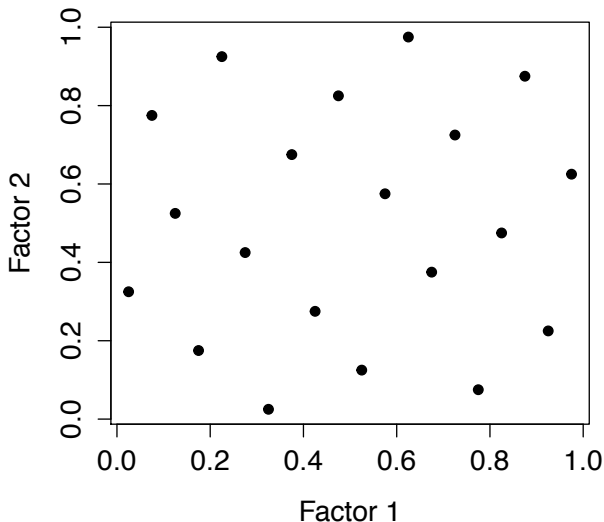
$$\frac{\partial \psi^P(D)}{\partial x_{rs}} = \frac{2}{\binom{n}{2}} \sum_{i \neq r} \frac{1}{\prod_{l=1}^P (x_{il} - x_{jl})^2} \frac{1}{(x_{is} - x_{js})}.$$

- R package: MaxPro.
- Available in JMP 12.

Maximum Projection Design (MaxPro) - Examples

MaxProLHD(20,2) ●

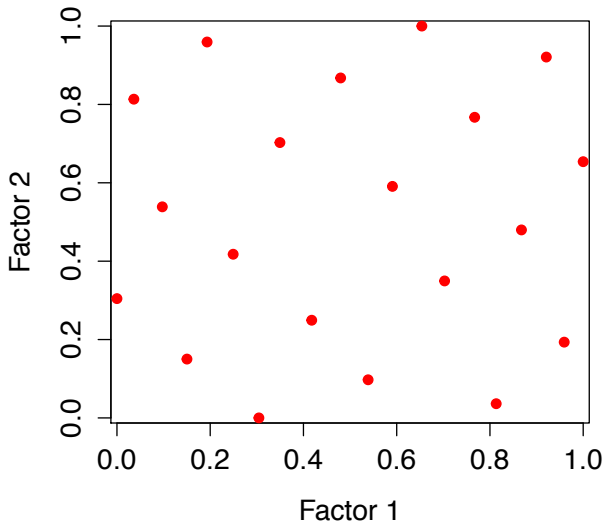
MaxPro(20,2) ●



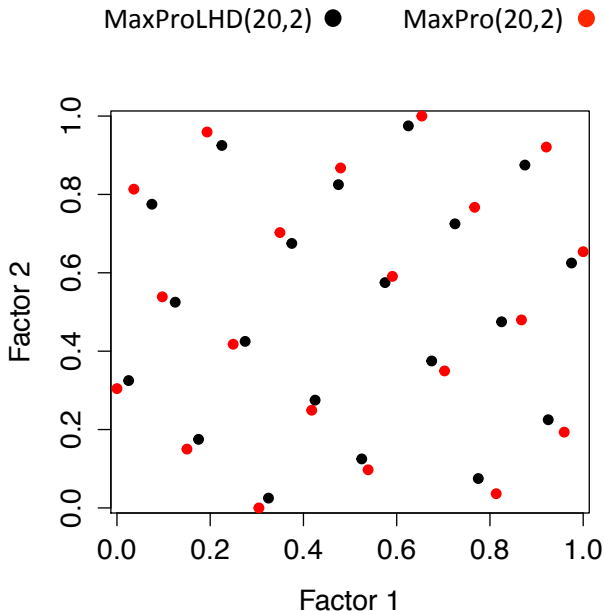
Maximum Projection Design (MaxPro) - Examples

MaxProLHD(20,2) ●

MaxPro(20,2) ●



Maximum Projection Design (MaxPro) - Examples

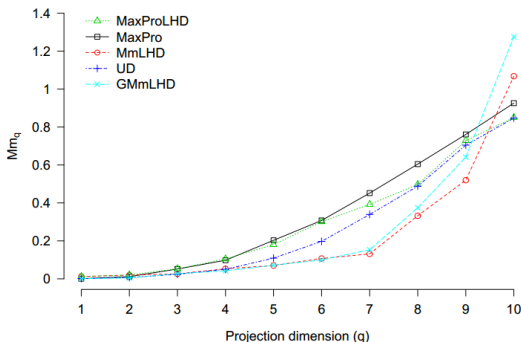


Maximum Projection Design (MaxPro) - Numerical Results

- Maximin Criterion:

$$Mm_q = \min_{r=1, \dots, \binom{p}{q}} \left(\frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{d_{qr}^{2q}(x_i, x_j)} \right)^{-1/(2q)}$$

Mm_q (larger-the-better)

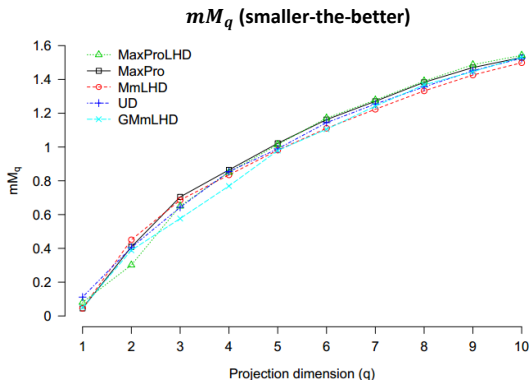


Maximum Projection Design (MaxPro) - Numerical Results

- miniMax Criterion:

$$mM_q = \max_{r=1,\dots,\binom{p}{q}} \max_{u \in \chi_q} \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{d_{qr}^{2q}(u, x_i)} \right)^{-1/(2q)},$$

where χ_q is the set of sample points.



Maximum Projection Design (MaxPro) - Properties

- Easy to compute MaxPro criterion.
- Good space-filling properties in projections to all subsets of factors.
- Available in R Package MaxPro and JMP 12.
- Advantageous in Gaussian process modeling.

Conclusion

- Latin hypercube design:
 - Good 1-dimensional projections, but might not be space-filling in p dimensions.
- miniMax/Maximin distance design:
 - Run size flexibility and good space-fillingness in p dimensions, but bad projections.
- Maximin Latin Hypercube design:
 - Good space-fillingness in p dimensions and uniform projections in a single dimension.
- Maximum projection design:
 - Good space-filling properties in projections to all subsets of factors.