1 Introduction

2 Modeling Computer Experiments

3 Designs for Computer Experiments

4 Uniform Projection Designs

5 Construction Methods

6 References
Introduction

What are computer experiments?

- Computer experiments are increasingly being used to explore the behavior of complex physical systems.
- A computer model is a large computer code that implements a complex mathematical model of a physical process. For example, simultaneous differential solver, finite element analysis, computational fluid dynamics.
A typical engineering model (Page 1 of 3)

\[
\begin{align*}
\rho_s A_s \frac{\partial^4 w}{\partial t^4} + E_s I_s \frac{\partial^4 w}{\partial x^4} &= \rho_s A_s \frac{\partial^4 w}{\partial t^4} + E_s I_s \frac{\partial^4 w}{\partial x^4} \\
&+ \{p, v, s \} \left[ \frac{\partial^3 u_s}{\partial t^3} + \frac{\partial^3 u_s}{\partial x^3} \right] + \{ \rho_s A_s \} \left[ \frac{\partial^3 u_s}{\partial t^3} + \frac{\partial^3 u_s}{\partial x^3} \right] \\
&+ \rho_s A_s \left[ \frac{\partial^3 u_s}{\partial x^3} + \frac{\partial^3 u_s}{\partial t^3} \right] + C_{11} I_s \left[ \frac{\partial^3 u_s}{\partial x^3} + \frac{\partial^3 u_s}{\partial t^3} \right] - E_s A_s \left[ \frac{\partial^3 u_s}{\partial x^3} + \frac{\partial^3 u_s}{\partial t^3} \right] + \beta \left[ \delta(x-x_1) - \delta(x-x_2) \right] \\
&+ \{ C_{11} I_s \} \left[ \frac{\partial^3 u_s}{\partial x^3} + \frac{\partial^3 u_s}{\partial t^3} \right] - E_s A_s \left[ \frac{\partial^3 u_s}{\partial x^3} + \frac{\partial^3 u_s}{\partial t^3} \right] + \beta \left[ \delta(x-x_1) - \delta(x-x_2) \right] = f(x,t)
\end{align*}
\]
Computer Experiments

Figure 1: Computer experiment
Computer Experiments

Characteristics of computer experiments

- Mostly deterministic (lack of random error)
- May take hours or even days to produce a single output
- Many input variables
- The performance of the predictor depends upon the choice of the training data (design).

Principles in traditional DOE are irrelevant

- Replication
- Blocking
- Randomization
Modeling Computer Experiments: Kriging

For $x \in \mathbb{R}^m$, treat the deterministic response $y(x)$ as a realization of a stochastic process

$$Y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x),$$

where $f_j(x)$ are known functions, $\beta_j$ are unknown parameters and $Z(\cdot)$ is a Gaussian process with mean 0 and covariance

$$\text{cov}(Z(w), Z(x)) = \sigma^2 R(w, x).$$

- This is the Kriging model used in spatial statistics.
- Also called Gaussian process model in Machine Learning.

R packages: DiceKriging, kergp, etc.
Prediction

Given a design $S = \{s_1, \ldots, s_n\}$ and data $y_S = (y(s_1), \ldots, y(s_n))^T$. Consider the linear predictor

$$\hat{y}(x) = c(x)^T y_S.$$  

Frequentists replace $y_S$ by the random vector $Y_S = \{Y(s_1), \ldots, Y(s_n)\}^T$, and compute the MSE.

The **Best Linear Unbiased Predictor** (BLUP): choose $c(x)$ to minimize

$$\text{MSE}[\hat{y}(x)] = E[c(x)^T Y_S - Y(x)]^2$$

subject to

$$E[\hat{y}(x)] = E[c(x)^T Y_S] = E[Y(x)]$$
Kriging model:  \( Y(x) = f(x)^T \beta + Z(x) \), where

\[
  f(x) = (f_1(x), \ldots, f_k(x))^T, \quad \beta = (\beta_1, \ldots, \beta_k)^T
\]

In matrix form:

\[
  Y_S = F \beta + Z, \quad \text{cov}(Z) = \sigma^2 R
\]

\[
  F = (f(s_1), \ldots, f(s_n))^T = (f_j(s_i))_{n \times k}
\]

\[
  R = (R(s_i, s_j))_{n \times n}
\]

\[
  r(x) = (R(s_1, x), \ldots, R(s_n, x))^T
\]

The generalized LS estimate and BLUP are

\[
  \hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y_S
\]

\[
  \hat{y}(x) = f(x)^T \hat{\beta} + r(x)^T R^{-1} (Y_S - F \hat{\beta})
\]

The GP interpolates the observed data: \( \hat{y}(s_i) = y(s_i) \) for \( s_i \in S \).
Correlation Functions

The correlation $R(w, x)$ has to be specified. Commonly used functions:

\[
R(w, x) = \prod \exp(-\theta_j |w_j - x_j|^{p_j}), \quad 0 < p_j \leq 2,
\]
\[
R(w, x) = \prod K(|w_j - x_j|; \theta_j)
\]

where $K()$ is Matérn correlation function with parameter $\nu = 5/2$.

\[
K(h; \theta) = \left(1 + \frac{\sqrt{5}h}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}h}{\theta}\right).
\]

The correlation parameters (e.g., $\theta_j$, $p_j$) need to be specified or estimated (by MLE or cross validation)

Given the correlation parameters, the MLEs are

\[
\hat{\beta} = \text{generalized l.s. estimate}
\]
\[
\hat{\sigma}^2 = \frac{1}{n}(Y_s - F\hat{\beta})' R^{-1} (Y_s - F\hat{\beta})
\]
Examples of Matérn $\nu = 5/2$ correlation functions
Data: \( y = \sin(2x)/(1 + x); \) Kriging: \( Y = \mu + Z(x). \)
A toy example: Kriging vs Polynomial models

Data: $y = \sin(2x)/(1 + x)$; Kriging: $Y = \mu + Z(x)$. 
Designs for Computer Experiments

- Constructing a “good” design is crucial for the success of a computer experiment.
- A “good” design should be **space-filling** (i.e., cover as much space as possible), and have **good projection properties**.
  - Latin hypercube designs (LHD) [McKay et al. (1979)]
  - Maximin and minimax distance designs [Johnson et al. (1990)]
  - Uniform designs [Fang et al. (2006)]
  - Maximum projection designs [Joseph et al. (2015)]
  - Uniform projection designs [Sun et al. (2019)]
- Optimality criteria: maximin distance, minimax distance, column-orthogonality, uniformity (discrepancy) etc.
- R packages: lhs, LHD, SLHD, UniDOE, MaxPro, etc.
Let $\hat{y}(x)$ be BLUE of $y(x)$ given a design $S = \{ s_1, \ldots, s_n \}$.

- **Integrated Mean Squared Error (IMSE)**

  $$\min_S \int_\mathcal{X} \text{MSE}[\hat{y}(x)] \phi(x) \, dx,$$

  where $\phi(x)$ is a given weight function.

- **Maximum Mean Squared Error (MMSE)**

  $$\min_S \max_{x \in \mathcal{X}} \text{MSE}[\hat{y}(x)].$$

- **Entropy (Gaussian process)**

  $$\max_S \det(R) = \det(R(s_i, s_j)).$$

- **Maximin distance criterion**:

  $$\max_S \min_{i < j} d(s_i, s_j)$$
Figure 2: Maximin LHD (left) and Minimax LHD (right) with $n = 7$ and $m = 2$
Maximin distance designs

For an \((n, s^m)\) design \(D = (x_{ik})_{n \times m}\),

\[d_p(x_i, x_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|^p,\]

Define the \(L_p\)-distance of \(D\) as

\[d_p(D) = \min\{d_p(x_i, x_j), 1 \leq i < j \leq n\}\]

Maximin distance design: maximize \(d_p(D)\) among all designs

Most constructions are based on stochastic algorithms:

- Morris and Mitchell (1995), Joseph and Hung (2008),
  Ba, Myers and Brenneman (2015, R package SLHD), etc.
- Flexible but are not effective for large designs.

Low-dimensional projections may not be space-filling.

- Saturated \(OA(n, 2^m)\)'s are maximin distance designs when \(m = n - 1\) (Xu 1999).
Uniform designs

Idea: choose design points from the design region with empirical distribution as “uniform” as possible (Fang et al, 2006).
For an $n \times m$ design $D$ over $[0, 1]^m$,

$$\text{Disc}(D) = \left\{ \int_{[0,1]^m} \left| \text{Vol}(J(a_x, x)) - \frac{N(D \cap J(a_x, x))}{n} \right|^2 \, dx \right\}^{1/2}.$$ 

The (squared) centered $L_2$-discrepancy is defined by

$$\text{CD}(D) = \left\{ \sum_{u \subseteq \{1:m\}} |\text{Disc}(D_u)|^2 \right\},$$

where $D_u$ is the projected design of $D$ onto dimensions indexed by the elements of $u$.

- Uniform designs may have poor projections in lower dimensional spaces.
Uniform Projection Designs: A New Class of Space-Filling Designs

- Focus on 2-dim projection uniformity

\[
\phi(D) = \frac{2}{m(m-1)} \sum_{|u|=2} CD(D_u),
\]  

(1)

- A design achieving the minimum \( \phi(D) \) value is a uniform projection design (UPD).
- The discrepancy has an analytical expression; for \( D = (z_{ik}) \) over \([0, 1]^m\):

\[
CD(D) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{k=1}^{m} \left( 1 + \frac{1}{2} |z_{ik}| + \frac{1}{2} |z_{jk}| - \frac{1}{2} |z_{ik} - z_{jk}| \right)
\]

\[
- \frac{2}{n} \sum_{i=1}^{n} \prod_{k=1}^{m} \left( 1 + \frac{1}{2} |z_{ik}| - \frac{1}{2} |z_{ik}|^2 \right) + \left( \frac{13}{12} \right)^m.
\]
### Why we need a new criterion? Four $25 \times 3$ LHDs

<table>
<thead>
<tr>
<th>Uniform $D_1$</th>
<th>Maximin $D_2$</th>
<th>MaxPro $D_3$</th>
<th>UPD $D_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 16 14</td>
<td>0 2 20</td>
<td>0 6 12</td>
<td>2 3 2</td>
</tr>
<tr>
<td>19 9 0</td>
<td>1 20 10</td>
<td>1 15 5</td>
<td>4 5 13</td>
</tr>
<tr>
<td>11 1 2</td>
<td>2 7 12</td>
<td>2 18 21</td>
<td>0 11 9</td>
</tr>
<tr>
<td>16 20 3</td>
<td>3 13 21</td>
<td>3 9 0</td>
<td>3 16 17</td>
</tr>
<tr>
<td>20 22 12</td>
<td>4 9 3</td>
<td>4 12 17</td>
<td>1 22 22</td>
</tr>
<tr>
<td>14 7 10</td>
<td>5 19 0</td>
<td>5 0 10</td>
<td>8 0 7</td>
</tr>
<tr>
<td>4 17 1</td>
<td>6 23 19</td>
<td>6 21 3</td>
<td>6 8 19</td>
</tr>
<tr>
<td>12 12 7</td>
<td>7 14 13</td>
<td>7 4 19</td>
<td>9 14 24</td>
</tr>
<tr>
<td>10 15 24</td>
<td>8 0 7</td>
<td>8 23 13</td>
<td>5 18 4</td>
</tr>
<tr>
<td>22 14 5</td>
<td>9 3 17</td>
<td>9 11 7</td>
<td>7 20 11</td>
</tr>
<tr>
<td>2 21 22</td>
<td>10 21 8</td>
<td>10 14 24</td>
<td>12 2 21</td>
</tr>
<tr>
<td>15 11 21</td>
<td>11 8 9</td>
<td>11 3 1</td>
<td>10 9 0</td>
</tr>
<tr>
<td>1 5 4</td>
<td>12 6 1</td>
<td>12 8 15</td>
<td>14 12 14</td>
</tr>
<tr>
<td>3 10 11</td>
<td>13 18 23</td>
<td>13 19 9</td>
<td>13 15 6</td>
</tr>
<tr>
<td>23 3 8</td>
<td>14 15 2</td>
<td>14 1 22</td>
<td>11 24 15</td>
</tr>
<tr>
<td>0 13 15</td>
<td>15 10 18</td>
<td>15 7 4</td>
<td>17 4 10</td>
</tr>
<tr>
<td>8 23 6</td>
<td>16 24 16</td>
<td>16 16 18</td>
<td>15 7 5</td>
</tr>
<tr>
<td>7 8 18</td>
<td>17 16 11</td>
<td>17 24 6</td>
<td>19 10 18</td>
</tr>
<tr>
<td>9 4 13</td>
<td>18 1 15</td>
<td>18 13 11</td>
<td>16 19 20</td>
</tr>
<tr>
<td>6 19 9</td>
<td>19 22 4</td>
<td>19 22 23</td>
<td>18 23 1</td>
</tr>
<tr>
<td>24 18 19</td>
<td>20 4 6</td>
<td>20 2 14</td>
<td>21 1 16</td>
</tr>
<tr>
<td>21 6 23</td>
<td>21 5 24</td>
<td>21 17 2</td>
<td>23 6 23</td>
</tr>
<tr>
<td>13 24 17</td>
<td>22 12 5</td>
<td>22 10 20</td>
<td>22 13 3</td>
</tr>
<tr>
<td>17 0 16</td>
<td>23 17 22</td>
<td>23 5 8</td>
<td>20 17 12</td>
</tr>
<tr>
<td>5 2 20</td>
<td>24 11 14</td>
<td>24 20 16</td>
<td>24 21 8</td>
</tr>
</tbody>
</table>
Bivariate projections of Uniform $D_1$ and Maximin $D_2$

Note: ‘X’ means that there are no points in the grid.
Bivariate projections of MaxPro $D_3$ and UPD $D_4$

Note: ‘X’ means that there are no points in the grid.
Some Theoretical Results

**Theorem 1**

For a balanced \((n, s^m)\) design \(D\) and any \(2 \leq k \leq m\),

\[
\frac{1}{\binom{m}{k}} \sum_{|u|=k} \phi(D_u) = \phi(D),
\]

where \(D_u\) is the projected design onto \(k\) factors indexed by \(u\).

- UPDs have good space-filling properties not only in two dimensions, but also in all dimensions.
Some Theoretical Results

**Theorem 2**

For a balanced \((n, s^m)\) design \(D = (x_{ik})\),

\[
\phi(D) = \frac{g(D)}{4m(m - 1)n^2s^2} + C(m, s),
\]

where

\[
g(D) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_1^2(x_i, x_j) - \frac{2}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{n} d_1(x_i, x_j) \right)^2
\]

- \(\phi(D)\) is a function of pairwise \(L_1\)-distances of the rows of \(D\).
- An equidistant design under the \(L_1\)-distance is a UPD.
Application: Design and Modeling Comparison

- A 512-run and 8-level full factorial design to study 3 drugs.
- The response was the ATP level of the cells after the drug treatments.
- Kriging model with noise: \( y(x) = \mu + Z(x) + \epsilon \)

<table>
<thead>
<tr>
<th></th>
<th>Normal Cell</th>
<th>Cancer Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D(_{512})</td>
<td>RD(_{80})</td>
</tr>
<tr>
<td>Kriging</td>
<td>0.002</td>
<td>0.21</td>
</tr>
<tr>
<td>NN</td>
<td>0.37</td>
<td>1.28</td>
</tr>
<tr>
<td>Polynomial</td>
<td>0.48</td>
<td>1.16</td>
</tr>
</tbody>
</table>

RD\(_{80}\): Random 80-run design; MPD\(_{25}\): MaxPro 25-run designs.
Comparison of projection properties

We compare four LHD(19, 18)'s:

1. The uniform design is from the uniform design website (UD).
2. The maximin distance design via R package SLHD (Ba, Myers and Brenneman, 2015, Technometrics).
3. The maximum projection (MaxPro) design were constructed via R package MaxPro (Joseph et al., 2015, Biometrika).
4. The uniform projection design (UPD): $E_b$.

We ran R commands `maximinSLHD` (with slice parameter $t = 1$) and `MaxProLHD` 100 times with default settings and chose the best designs.
Comparison of projection properties

Four criteria will be used in the comparison:

1. minimum Euclidean distance
2. maximum projection criterion (Joseph et al. 2015)
   \[
   \psi(D) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{k=1}^{m} (x_{ik} - x_{jk})^2} \right\}^{1/m}
   \]
3. relative maximum centered $L_2$-discrepancy (CD)
4. maximum correlation $\rho_{ave}$.

For each $k$, we evaluate all $\binom{m}{k}$ projected designs and determine the worst projection with respect to four criteria.
Figure 3: (a) minimum Euclidean distance (the larger the better), (b) maximum $\psi(D)$ (the smaller the better), (c) relative maximum CD (the smaller the better), and (d) maximum $\rho_{\text{ave}}$ (the smaller the better).
Construction Methods

- Good Lattice Point (GLP) designs are LHDs and often used to construct uniform designs (Fang and Wang, 1994).
- Let $h_1 < \ldots < h_p$ be $p$ integers (from 1 to $n$) coprime to $n$

$$D = (x_{ij}) \text{ with } x_{ij} = i \times h_j \pmod{n}$$

An example $n = 7$:

$$D = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 4 & 6 & 1 & 3 & 5 \\
3 & 6 & 2 & 5 & 1 & 4 \\
4 & 1 & 5 & 2 & 6 & 3 \\
5 & 3 & 1 & 6 & 4 & 2 \\
6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \quad \text{with } d(D) = 12 \quad \text{(while } d_{\text{upper}} = 16)$$
GLP designs

Results from Zhou and Xu (2015, Biometrika)

- An upper bound (for $L_1$-distance): For any $N \times n$ LHD $D$,

$$d(D) \leq d_{\text{upper}} = \lfloor (N + 1)n/3 \rfloor,$$

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

- Obtain the distances for four classes of GLP designs.
  - For an odd prime $n$, the $n \times (n - 1)$ GLP design has
    $$d(D) = (n + 1)(n - 1)/4.$$
  - the upper bound is $d_{\text{upper}} = (n + 1)(n - 1)/3$

- The $d_{\text{eff}}(D) = d(D)/d_{\text{upper}}$ for a GLP design is 75%.

- A surprising result: any linear level permutation of any column does not decrease the distance $d(D)$. 

GLP + Linear Permutation

Example: \( n = 7 \): Total \( 7^6 = 117,649 \) linear permutations.

- consider only 7 simple permutations: \( D_i = D + i \mod n \)

\[
\begin{pmatrix}
D \\
D_1
\end{pmatrix} \rightarrow \begin{pmatrix}
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 4 & 6 & 1 & 3 & 5 \\
3 & 6 & 2 & 5 & 1 & 4 \\
4 & 1 & 5 & 2 & 6 & 3 \\
5 & 3 & 1 & 6 & 4 & 2 \\
6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \\
\begin{pmatrix}
2 & 3 & 4 & 5 & 6 & 0 \\
3 & 5 & 0 & 2 & 4 & 6 \\
4 & 0 & 3 & 6 & 2 & 5 \\
5 & 2 & 6 & 3 & 0 & 4 \\
6 & 4 & 2 & 0 & 5 & 3 \\
0 & 6 & 5 & 4 & 3 & 2 \\
1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\end{pmatrix}
\]

\( d(D) = 12 \quad d(D_1) = 13 \)

- After linear permutations, \( d_{\text{eff}} \) is about 90% for large \( n \).
How About Nonlinear Permutations?

- Given an integer $n$, for $x = 0, \ldots, n - 1$,

$$W(x) = \begin{cases} 
2x, & \text{for } 0 \leq x < n/2; \\
2(n - x) - 1, & \text{for } n/2 \leq x < n.
\end{cases}$$

- The $W$ is a permutation of $\{0, \ldots, n - 1\}$.

The $W$ has been useful in

1. Latin squares, Williams (1949)
3. Orthogonal LHDs under a second-order Fourier model, Butler (2001)
GLP + Williams Transformation

Algorithm (Wang, Xiao and Xu, 2018, Annals of Statistics)

**Step 1.** Generate an $n \times p$ GLP design $D$.

**Step 2.** For $b = 0, \ldots, n - 1$, generate $D_b = D + b \pmod{n}$.

**Step 3.** Let $E_b = W(D_b)$.

**Step 4.** Find the best $E_b$ which maximizes $d(E_b)$.

**Example:** $n = 7$, $b = 1$

$$D \quad \rightarrow \quad D_1 = D + 1 \pmod{n} \quad \rightarrow \quad E_1 = W(D_1)$$

$$
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 4 & 6 & 1 & 3 & 5 \\
3 & 6 & 2 & 5 & 1 & 4 \\
4 & 1 & 5 & 2 & 6 & 3 \\
5 & 3 & 1 & 6 & 4 & 2 \\
6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\quad \rightarrow 
\begin{pmatrix}
2 & 3 & 4 & 5 & 6 & 0 \\
3 & 5 & 0 & 2 & 4 & 6 \\
4 & 0 & 3 & 6 & 2 & 5 \\
5 & 2 & 6 & 3 & 0 & 4 \\
6 & 4 & 2 & 0 & 5 & 3 \\
0 & 6 & 5 & 4 & 3 & 2 \\
1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\rightarrow
\begin{pmatrix}
4 & 6 & 5 & 3 & 1 & 0 \\
6 & 3 & 0 & 4 & 5 & 1 \\
5 & 0 & 6 & 1 & 4 & 3 \\
3 & 4 & 1 & 6 & 0 & 5 \\
1 & 5 & 4 & 0 & 3 & 6 \\
0 & 1 & 3 & 5 & 6 & 4 \\
2 & 2 & 2 & 2 & 2 & 2
\end{pmatrix}
$$

$d(D) = 12 \quad \quad d(D_1) = 13 \quad \quad d(E_1) = 16 \quad (=d_{upper})$
Comparison of Various $n \times (n - 1)$ LHDs

![Graph showing comparison of various LHDs with different symbols and colors.](image-url)

Legend:
- Red: GLP+WT
- Green: Zhou and Xu 2015
- Blue: Xiao and Xu 2017 I
- Black: Xiao and Xu 2017 II
- Orange: GLP
- Purple: SLHD
Key Result

Let

\[ b = W^{-1} \left( \frac{n - 1}{2} \pm c \right) \]

where \( c = \lfloor \sqrt{(n^2 - 1)/12} \rfloor \).

**Theorem 3**

*Given a prime \( n \) and \( p = n - 1 \), such defined \( b \) leads the best \( E_b \), with*

\[ d_{\text{eff}}(E_b) \geq 1 - 2/\sqrt{3(n^2 - 1)}. \]

*As \( n \to \infty \), \( d_{\text{eff}}(E_b) \to 1 \).*

- No need for computer search: \( D \to D_b \to E_b = W(D_b) \)
- Guaranteed efficiency
- Larger \( n \), better design
For any $n \times p$ design $D = (x_{ij})$, define

$$
\rho_{\text{ave}}(D) = \frac{\sum_{j \neq k} |\rho_{jk}|}{p(p-1)},
$$

where $\rho_{jk}$ is the correlation between columns $j$ and $k$ of $D$. 
Comparison of $\rho_{ave}$ for $n \times (n - 1)$ Designs

Method I: $\rho_{ave}(E_b) < 2/(n - 2)$
Many available algorithms for constructing space-filling designs, but not efficient for constructing large designs

A breakthrough — Wang, Xiao and Xu (2018) constructed maximin distance designs via good lattice points and a nonlinear transformation without computer search.

- Large distance efficiencies: $d_{eff} = 1$ (or $\to 1$)
- Low average correlation $\rho_{ave} \to 0$ as $n \to \infty$
- Asymptotically optimal under the uniform projection criterion.

A new class of space-filling designs — Uniform projection designs

- suitable when only a subset of the input variables are active.
- good space-filling not only in two dimensions, but also in all dimensions.
- equivalent to maximin $L_1$-distance criterion if $L_1$-equidistant designs exist.
- robust under other design criteria.

There are still many open problems to be investigated.
Selected References


