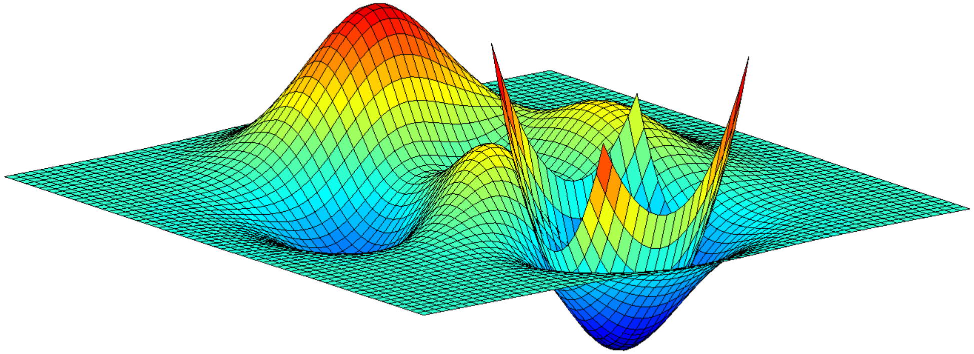
Surrogate Models and Design by Experiments

TMKT48 Design Optimization

LINKÖPING UNIVERSITY

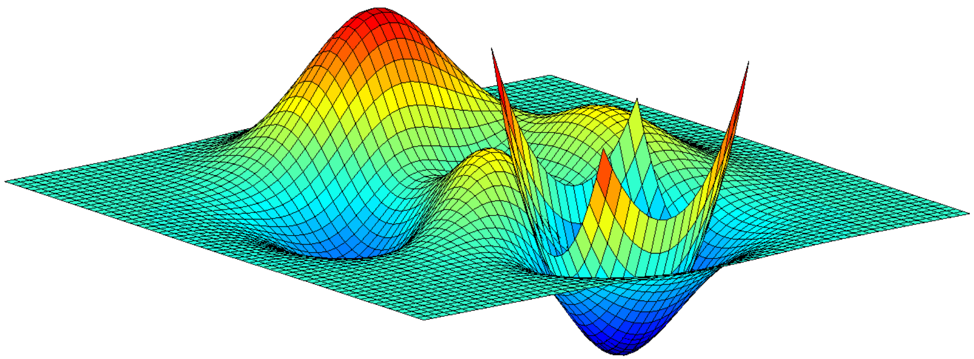
Content

- What is a surrogate model?
- How to use surrogate models
- Types of surrogate models
- Design of Experiments
  - How to choose which experiments to perform



Surrogate Models

- Also known as metamodels (models of models)
- Numerically efficient reanimations of systems or other models
- Are used to model unknown systems
- Can replace computationally expensive models to enable optimization

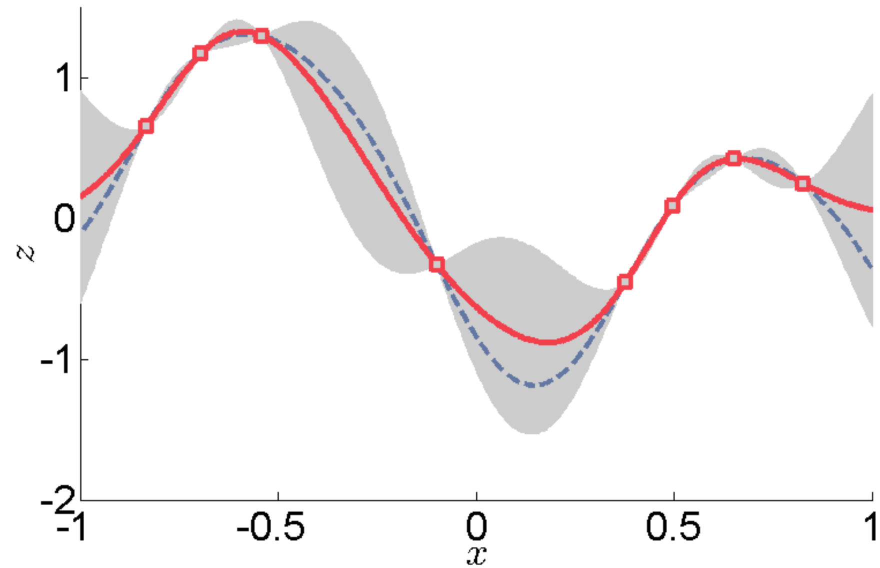


How to use surrogate models

- Collect the data needed
  - Experiments / Simulation
- Create a model that reanimates the data from the experiments
- Perform an optimization of the surrogate model to find an optimal design
- Verify the optimal design
  - Experiment / Simulation

Common Types of Surrogate Models

- Polynomial Response Surfaces
- Neural Networks
- Kriging
- Radial Basis Functions
- Support Vector Regression



Wikipedia: Kriging

Polynomial Response Surfaces

- Approximates the desired entity as a polynomial of desired degree

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 = \mathbf{X\beta}$$

- Can be fitted with the linear least squares method

$$\mathbf{X\beta} = \mathbf{y}$$

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

Polynomial Response Surfaces

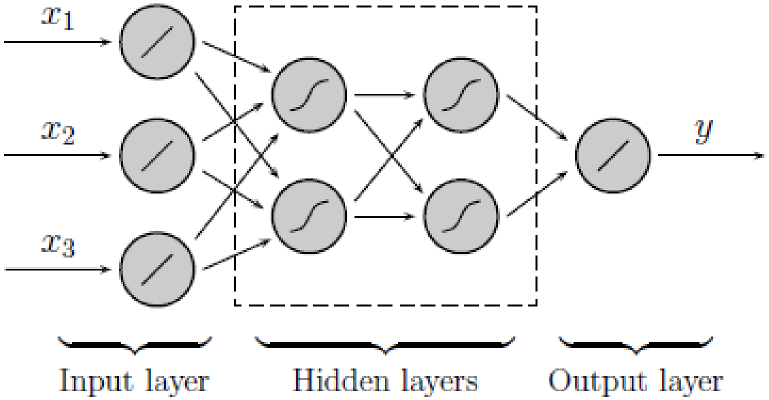
- Approximates the desired entity as a polynomial of desired degree

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 = \mathbf{X\beta}$$

- Pros
  - Easy to implement and understand
  - Computationally fast creation (matrix problem)
- Cons
  - Unsuitable for problems with many parameters
    - Too many samples are needed

Neural Networks

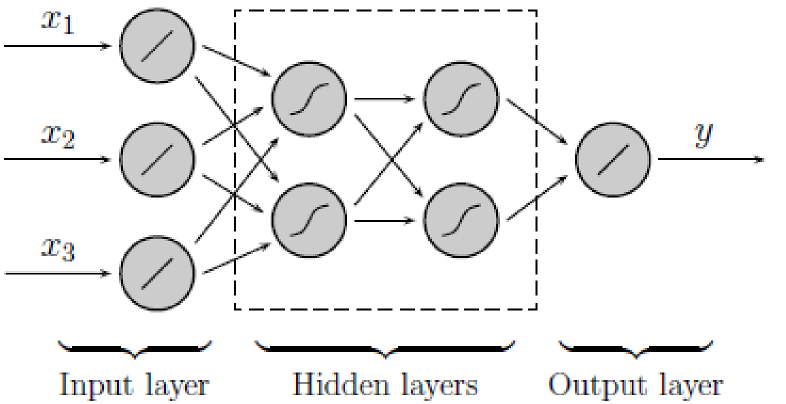
- Reanimates the way the human brain processes information
- A NN consists of several layers – an input layer, an output layer and one or more hidden layers.
- The variables in the layers are called nodes and a node uses a combination of the outputs of the nodes from the previous layers.



From David Lönn's dissertation

Neural Networks

- It is often used to duplicate data
- Pros
  - Highly flexible
- Cons
  - It is very time consuming to create



From David Lönn's dissertation

Interpolation Techniques

- The estimation is a linear combination of the value at known points.

- Often uses the distance to known points to calculate the weights

$$\hat{y} = \sum_{j=1}^{N_c} \omega_j y(x_j)$$

$$\omega_j = \frac{\frac{1}{d_j}}{\sum_{i=1}^{N_c} d_i}$$

$$\hat{y} = \frac{\sum_{j=1}^{N_c} \frac{1}{d_j} y(x_j)}{\sum_{j=1}^{N_c} \frac{1}{d_j}}$$

Radial Basis Functions

- Interpolation technique
- Uses the Euclidian distance to known points to estimate the new point

$$\hat{y} = \sum_{j=1}^{N_c} \omega_j \phi_j(x)$$

$$\phi_j(x) = \Phi\left(\left\|x - x_j\right\|\right)$$

$$\begin{pmatrix} \phi_1(x_1) & \cdots & \phi_{N_c}(x_1) \\ \phi_1(x_2) & \cdots & \phi_{N_c}(x_2) \\ \vdots & \vdots & \vdots \\ \phi_1(x_{N_c}) & \cdots & \phi_{N_c}(x_{N_c}) \end{pmatrix} \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{N_c} \end{Bmatrix} = \begin{Bmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_{N_c}) \end{Bmatrix}$$

Radial Basis Functions

- Interpolation technique

$$\hat{y} = \sum_{j=1}^{N_c} \omega_j \phi_j(x)$$

- Uses the Euclidian distance to known points to estimate the new point

$$\phi_j(x) = \Phi\left(\left\|x - x_j\right\|\right)$$

- Example of Φ: Gaussian function:

$$\Phi(r) = e^{-\frac{r^2}{s^2}}$$

$$s \approx d_{\max}\left(nN_c\right)^{-\frac{1}{n}}$$

LINKÖPING UNIVERSITY

Radial Basis Functions

- Interpolation technique
- Uses the Euclidian distance to known points to estimate the new point
- Possibility to add global trends
  - low order polynomial
- Pros
  - Quite good accuracy
- Cons
  - Requires optimization of the parameters to achieve optimal accuracy

Kriging

- Interpolating
- Used in geostatistics

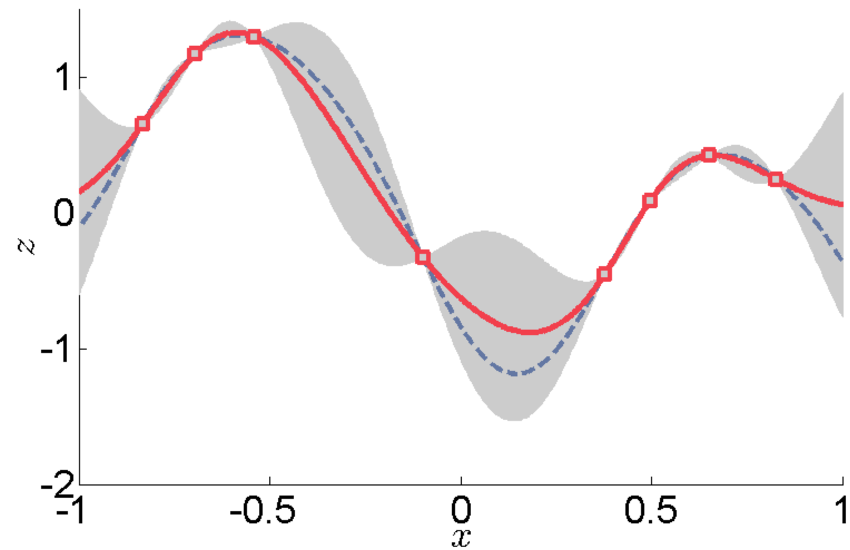
$$\hat{y} = \sum_{j=1}^{N_c} \omega_j y(x_j) \quad \begin{pmatrix} 0 & \gamma(h_{12}) & \cdots & \gamma(h_{1N_c}) \\ \gamma(h_{21}) & 0 & \cdots & \gamma(h_{2N_c}) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma(h_{N_c1}) & \gamma(h_{N_c2}) & \cdots & 0 \end{pmatrix} \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{N_c} \end{Bmatrix} = \begin{Bmatrix} \gamma(h_{p1}) \\ \gamma(h_{p2}) \\ \vdots \\ \gamma(h_{pN_c}) \end{Bmatrix}$$

- Example

$$\gamma(h) = C_0 + C\left(1 - e^{-\frac{h}{a}}\right)$$

Kriging

- Pros
  - Great accuracy
  - Gives an estimation of the prediction error
- Cons
  - The fitting process takes time since it is an optimization problem
  - Bad if some samples are close to each other



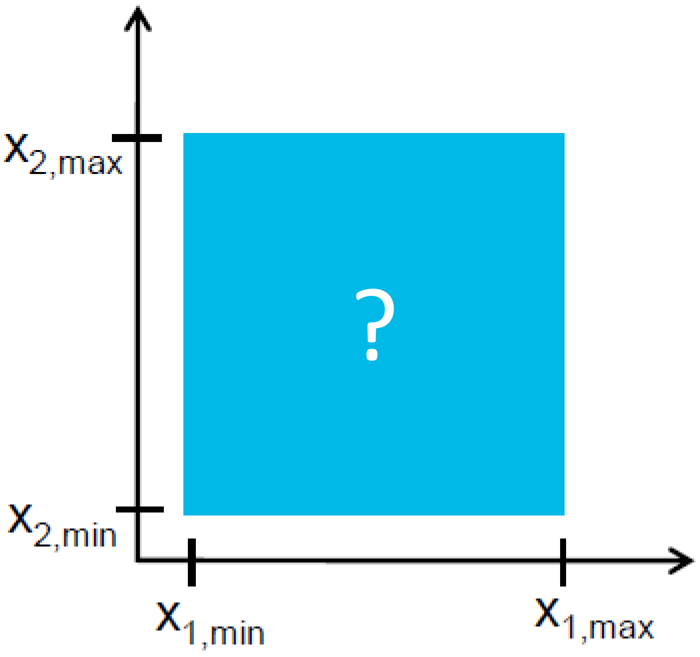
Design of Experiments

Which experiments should we perform to get as good surrogate models as possible with as few experiments as possible?

Design of Experiments

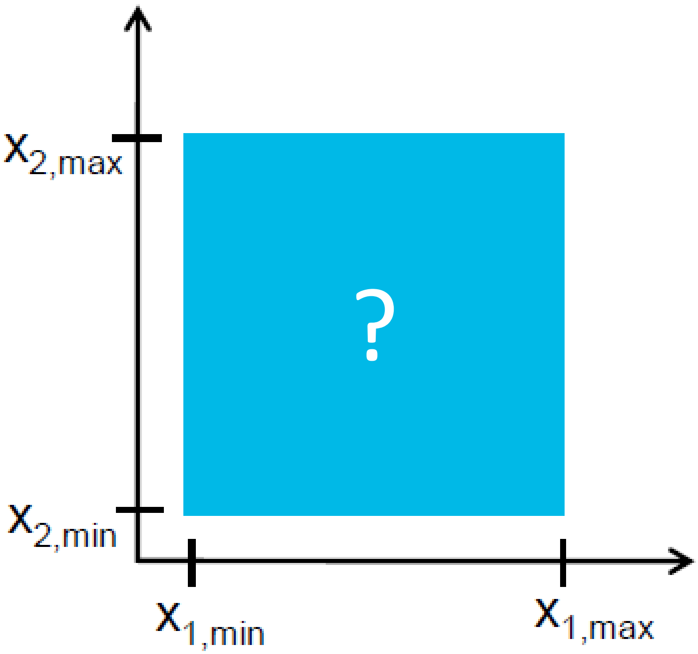
- Answers the question:

- Which experiments should we perform to get as good surrogate models as possible with as few experiments as possible?



Common Types of DoE

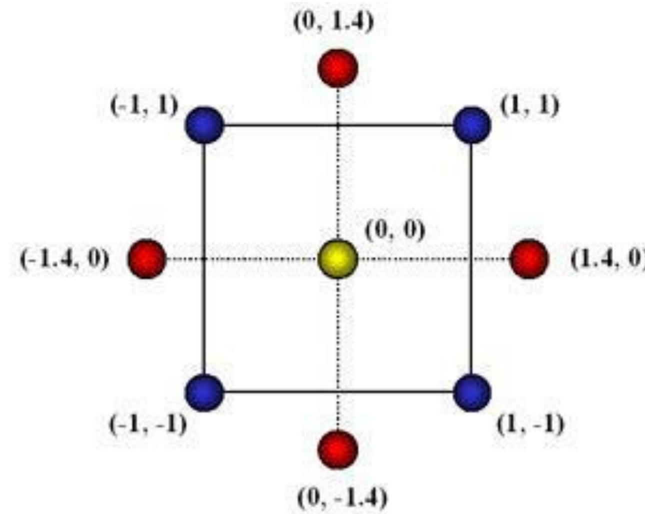
- 2-Level Factorial Designs
- Central Composite Design
- D-Optimal
- Latin Hypercube Sampling
- Maximum Entropy Design

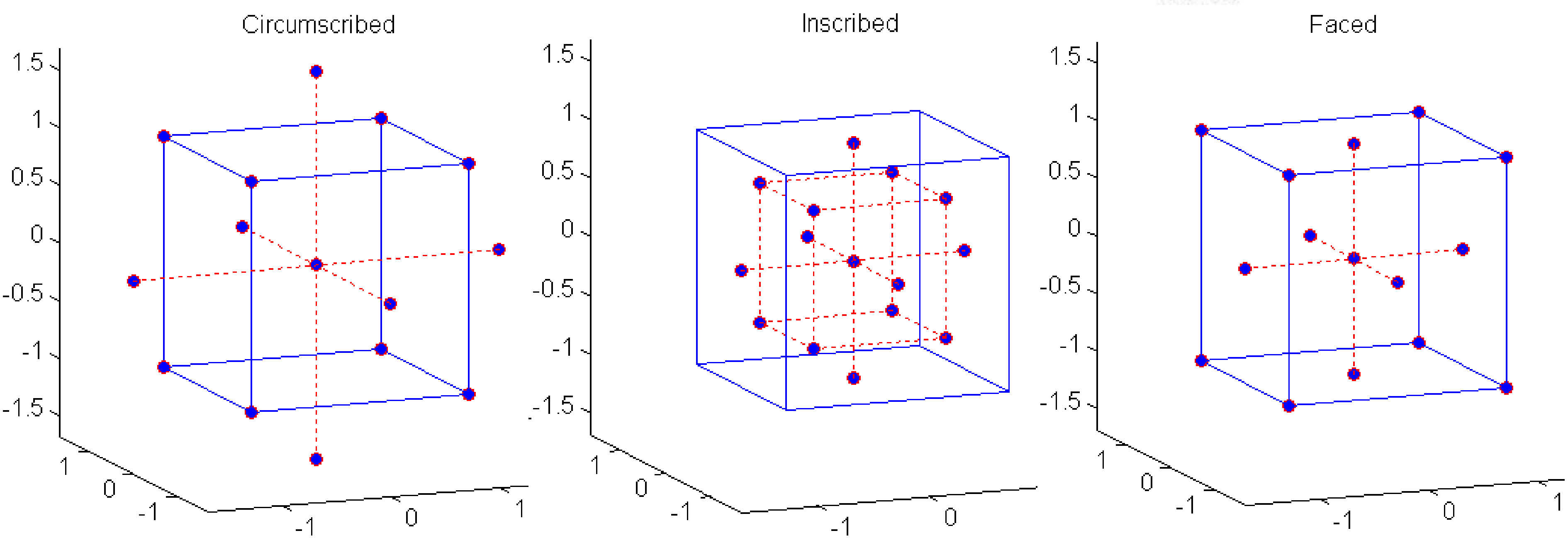


LINKÖPING UNIVERSITY

Central Composite Design

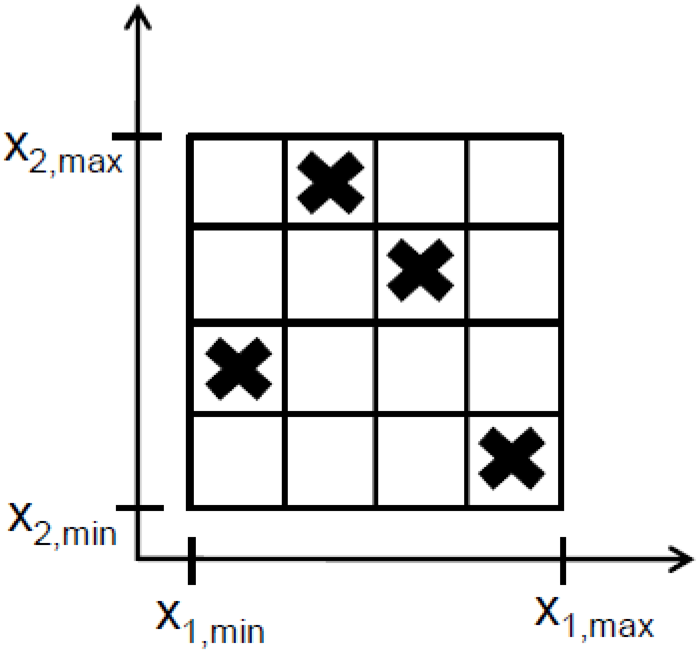
- Central Sample
- One Sample for each axis
- One sample for each corner



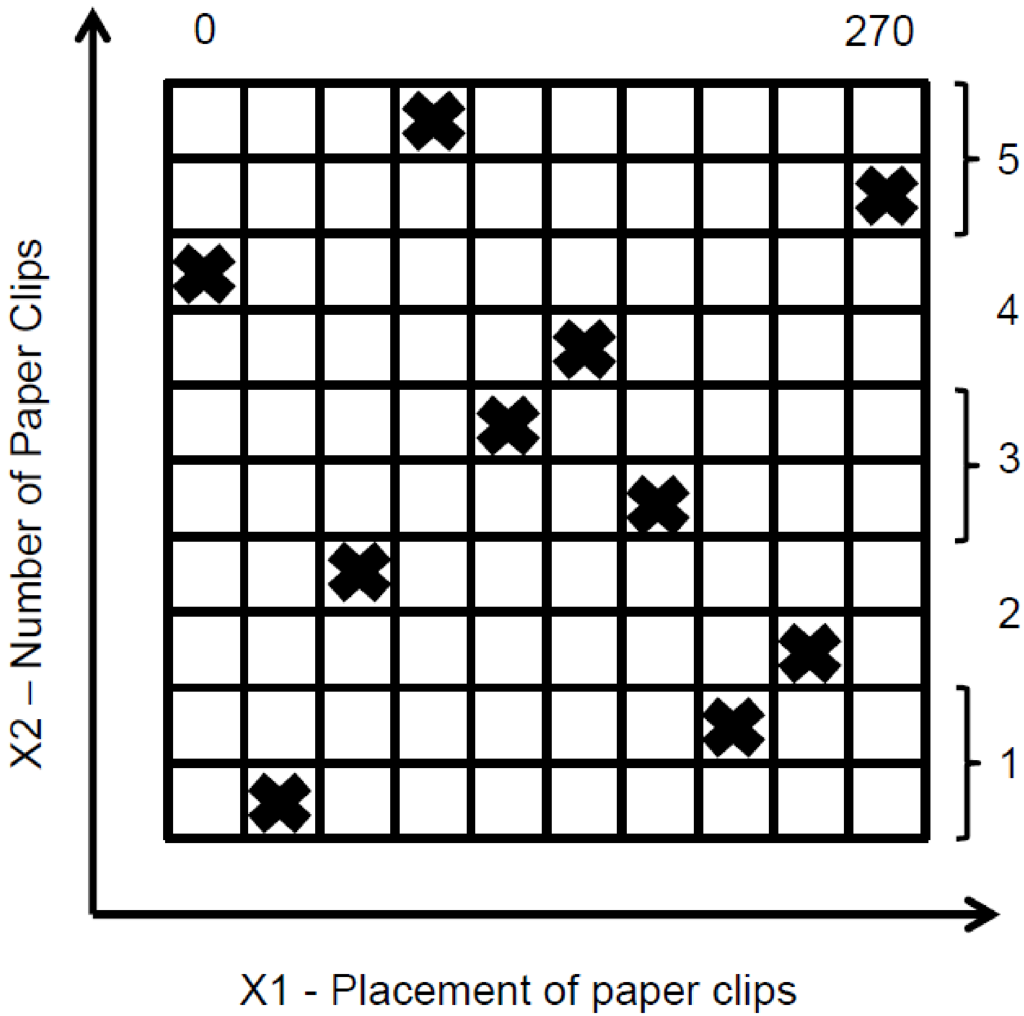


Latin Hypercube Sampling

- Divides the design space into intervals
- Draw one sample from each row and column



Latin Hypercube Sampling - Example



X1	X2
0	4
30	1
60	2
90	5
120	3
150	4
180	3
210	1
240	2
270	5

Questions?