# Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling

Akira Horiguchi

The Ohio State University
Computer Experiments Reading Group: STAT 8010.02

Thursday, March 29, 2018

# Introduction

# About the Paper

*Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling* (2011) by K. Crombecq, E. Laermans, T. Dhaene.

- Comparison and analysis of different space-filling sequential design methods
  - Three novel methods created by authors
  - Several other state-of-the-art methods from other authors
- All methods compared on a set of examples
- Advantages and disadvantages discussed

# Low-level introduction

Ford Motor Company car crash simulator

- 36 to 160 hours for a single instance

Important to make simulators **faster**

# Assumptions

Simulation assumptions:

1. System under study is a black box
2. Simulator is deterministic
   - Determinisitic noise

# Global surrogate modeling

Loosely,

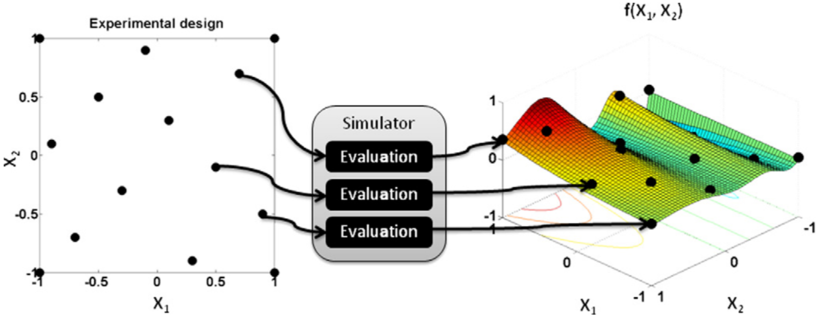
- Find approximation function $\tilde{f}$ that
  1. mimics $f$
  2. can be evaluated much faster than $f$

Mathematically,

- Simulator: unknown function $f : \mathbb{R}^d \to \mathbb{C}$
- $f$ is sampled at $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\} \subset [-1, 1]^d$
  - Function values $\{f(\mathbf{p}_1), f(\mathbf{p}_2), \ldots, f(\mathbf{p}_n)\}$ are known
- Choose $\tilde{f} : \mathbb{R}^d \to \mathbb{C}$ from possibly infinite set of candidate approximation functions

(Write down $f$, $\tilde{f}$, $P$)

# Global surrogate modeling



# Experimental Design

How to choose data points $P$ (aka experimental **design**)?

- Important to success of surrogate modeling task
- Choose data points that capture most information about $f$
  - Difficult! Little is known about $f$ in advance

# Table of Contents



**Efficient** *space-filling and non-collapsing sequential* **design strategies for simulation-based modeling**

# Sequential design

# Why sequential design?

Traditional design of experiments (DoE)

1. Choose $P$ based only on info available before first simulation
2. Feed $P$ to simulator
3. Build $\tilde{f}$

# Why sequential design?

Deterministic computer experiments

- Replication, randomization, and blocking lose their relevance
- Leaves space-filling designs as the only interesting option
    - Cover domain as equally as possible

# Why sequential design?

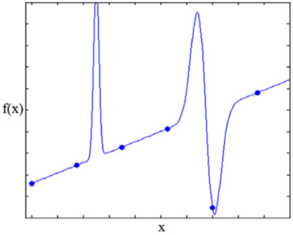
**Sequential** design (aka adaptive sampling)

- Transforms "one-shot" traditional algorithm into iterative process

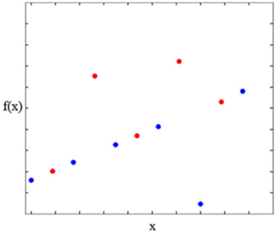
Why iterate?

- Sequentially gain more information about $f$ before choosing next design points
    - Explore more interesting areas
    - Allocate design points to difficult-to-approximate areas
- No need to choose no. design points ahead of time

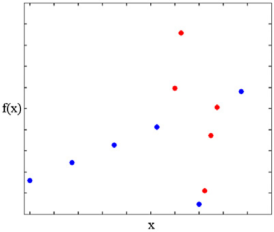
# Why sequential design?




(a) Initial set of samples




(b) Exploration




(c) Exploitation

# Important criteria for experimental designs

# What makes a good experimental design?

1. Granularity
2. Space-filling
3. Non-collapsing (good projective properties)

# Granularity

Granularity of a strategy

- Refers to number of points selected during each iteration of algorithm
- Coarse-grained sequential design strategy
  - Large number of points selected
- Fine-grained sequential design strategy
  - Small (preferably one) number of points selected

# Granularity

Why is fine-grained prefered?

- Avoids over- or undersampling
  - Don't know ahead of time how many design points to pick
- Computation time might run out!
  - Punch card days

# Space-filling

What is a **space-filling** design?

- Intuitively, points are spread out evenly over design space
- Mathematically, select design $P$ to maximize criterion
  - Several space-filling criteria have been proposed
    - E.g. Manhattan, Maximin, Audze-Eglais, Centered $L_2$ discrepancy, $\phi_p$
  - Choose one (or combination) of criteria
  - Maximin space-filling criterion used in this paper

# Space-filling

What is a maximin space-filling criterion?

- Maximize smallest $L_2$ distance between any two points in design
  - I.e. maximize $min_{\mathbf{p_i},\mathbf{p_j} \in P} ||\mathbf{p_i} - \mathbf{p_j}||_2$

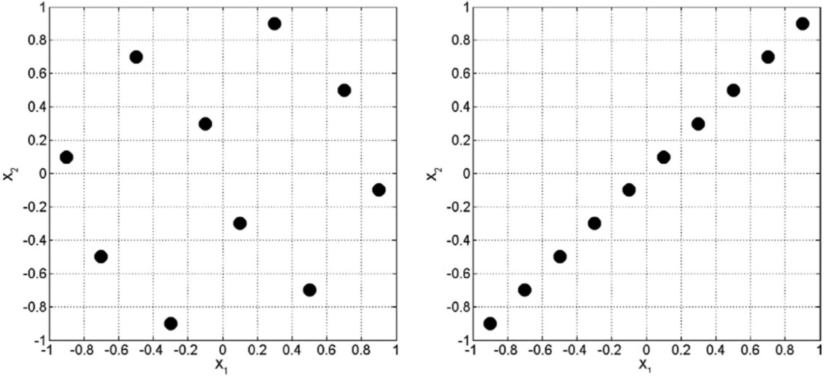
From now on, $min_{\mathbf{p_i},\mathbf{p_j} \in P} ||\mathbf{p_i} - \mathbf{p_j}||_2$ refered to as *intersite distance*

# Non-collapsing

What is a design that has good projective properties? (Also called the **non-collapsing** property.)

- When design is projected from $d$-dim space to $(d-1)$-dim space along one of the axes, no two points are ever projected onto each other
  - I.e. for every point $\mathbf{p}_i$, each value of $p_i^k$ is strictly unique

# Non-collapsing



# Existing methods

# Some existing methods

To be used as benchmarks:

1. Factorial designs
2. Latin hypercube
3. Low-discrepancy sequences
4. Remaining methods

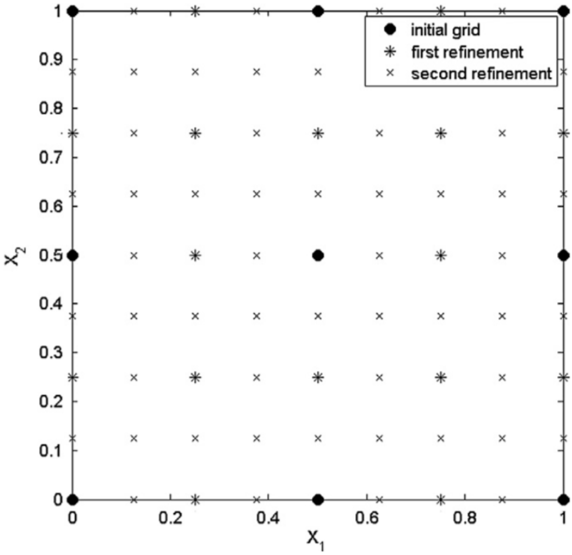
Design space is hypercube $[-1, 1]^d$

# Factorial designs

What is a full factorial design (`factorial`)?

- Construction
  - Grid of $m^d$ points
- Automatic advantages
  - Largest intersite distance among all designs
- Disadvantages
  - Horrible projective properties

# Factorial designs

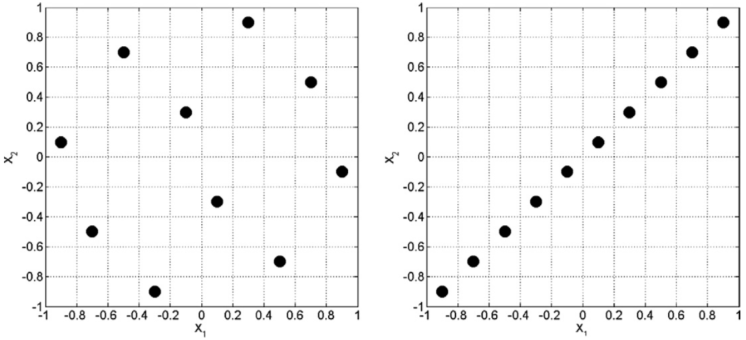


# Latin hypercube

What is a Latin hypercube design (LHD)?

- Construction
  - Divide each dimension in $m$ equally sized intervals
  - Place exactly one point in each interval for each dimension
- Automatic advantages
  - Largest projective distance among all methods
  - Any two points are at least $\frac{2}{m}\sqrt{2}$ distance away
- Achtung!
  - Can have bad space-filling properties
  - Constructing a good space-filling LHD is non-trivial
    - Can take 100+ hours in $d = 3$ setting
- Three LHD generation methods used
  - `lhd-joseph`
  - `lhd-matlab`
  - `lhd-optimal` (available for certain combos of dims and pts)

# Latin hypercube




(a) Optimal Latin hypercube

(b) Bad Latin hypercube

# Low-discrepancy sequences

What does low-discrepancy mean?

- A set of points $P$ has a low discrepancy if the number of points from the dataset falling into an arbitrary subset of the design space is close to proportional to a particular measure of size for this subset

# Low-discrepancy sequences

What is a low-discrepancy sequence?

- Sequences of points such that for each $n$, the points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ have a low discrepancy
- Advantages
  - Popular sequences have good projective properties
- Disadvantages
  - For small $n$, bad space-filling properties
- Two low-discrepancy sequences used
  - `Halton`
  - `Sobol`

# Remaining methods

Three other methods to be used

- Methods from Crombecq et al. (2009)
  1. `delaunay`
     1. Computes delaunay triangulation of samples
     2. Selects new sample in center of gravity of simplex with largest volume
  2. `voronoi`
     1. Estimates Voronoi tessellation of samples
     2. Selects new sample in largest Voronoi cell
  3. `random` sampling
     - Base case
- Fine-grained
- Optimize toward intersite distance
- Neglect projective distance

# New space-filling sequential design methods

# Introduction

Goal:

- Score well on space-filling and non-collapsing criteria
- Fine-grained as possible

# Introduction

New methods

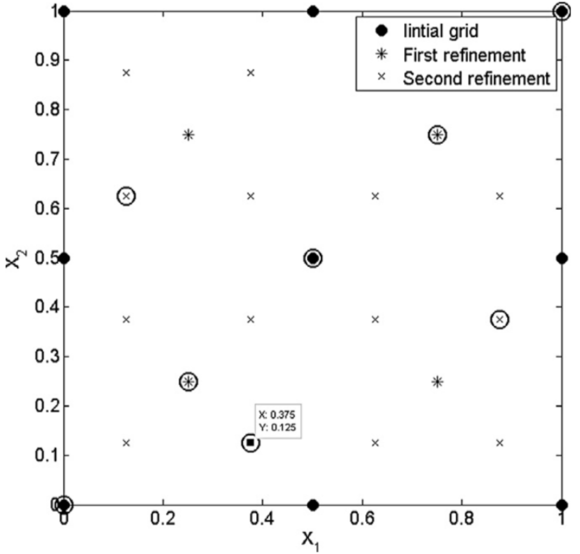
1. Sequential nested Latin hypercubes
2. Global Monte Carlo methods
3. Optimization-based methods

# Sequential nested Latin hypercubes

How to "sequentialize" LHD (`lhd-nested`)?

- Repeat:
  1. Grid of candidate (initially $m^d$) points
  2. Iteratively choose new samples (initially $m$) on grid
     - Chosen point lies farthest away from all previously selected points

# Sequential nested Latin hypercubes



# Global Monte Carlo methods

Monte Carlo methods in sequential design

1. Generate large number of random candidate points
2. Compute criterion for all these points
3. Select point with the highest score on criterion

# Global Monte Carlo methods

First MC criterion used: `mc-intersite-proj`

- Aggregate of intersite and projected distance
- Want to score candidate design $P' = P \cup \mathbf{p}$
  - $P$ is previously evaluated samples
  - $\mathbf{p}$ is new candidate point
- Score of $P'$ is

$$\mathtt{intersite} - \mathtt{proj}(P, \mathbf{p}) = \frac{\sqrt[d]{n+1} - 1}{2} \min_{\mathbf{p_i} \in P} ||\mathbf{p_i} - \mathbf{p}||_2 + \frac{n+1}{2} \min_{\mathbf{p_i} \in P} ||\mathbf{p_i} - \mathbf{p}||_{-\infty}$$

# Global Monte Carlo methods

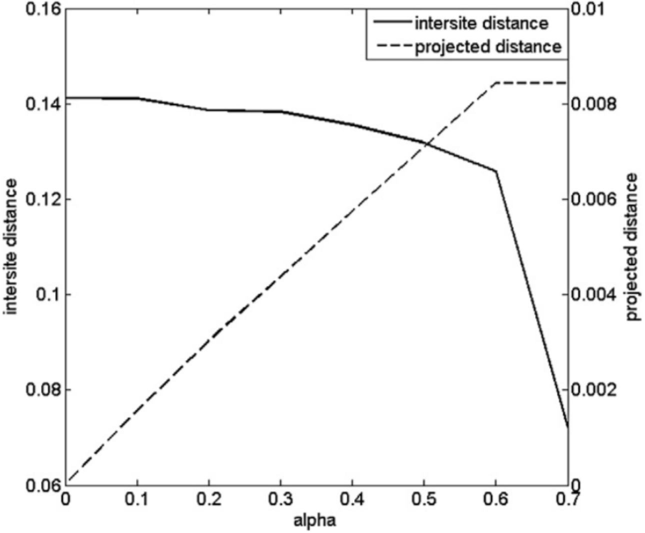
Second MC criterion used: `mc-intersite-proj-th`

- Still use intersite and projected distance
- Instead, use projected distance as threshold function
  - Discard points that lie too close (projected) to other points
- Threshold (minimum allowed projected distance) is $d_{min} = \frac{2\alpha}{n}$
  - $\alpha$ is tolerance parameter
- Score of $P'$ is

$$\mathtt{intersite} - \mathtt{proj} - \mathtt{th}(P, \mathbf{p}) = \min_{\mathbf{p_i} \in P} ||\mathbf{p_i} - \mathbf{p}||_2 \times \mathbf{1}_{\{\min_{\mathbf{p_i} \in P} ||\mathbf{p_i} - \mathbf{p}||_{-\infty} \geq d_{min}\}}$$

- $\alpha = 0.5$ chosen (tradeoff)

# Global Monte Carlo methods



# Optimization-based methods

First optimization-based criterion used: `optimizer-proj`

1. Find 30 points with large minimum intersite distance
2. Wiggle points to maximize minimum projected distance ($\beta = 0.3$ chosen)
3. Select point with largest minimum projected distance

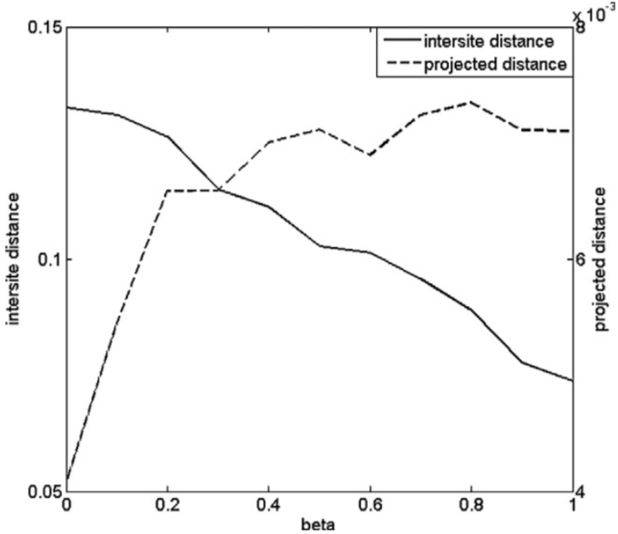
---

**Algorithm 1.** The optimizer-proj algorithm

---

$P_{candidates} \leftarrow 100n$ random points
$P_{new} \leftarrow$ 30 best points using `intersite distance`
**for all** $\mathbf{p_{new}} \in P_{new}$ **do**
 $m(\mathbf{p_{new}}) \leftarrow min_{\mathbf{p} \in P} \| \mathbf{p_{new}} - \mathbf{p} \|_2$
 $d_{max} \leftarrow \frac{\beta m(\mathbf{p_{new}})}{2}$
 Optimize $\mathbf{p_{new}}$ towards $\|P \cup \mathbf{p_{new}}\|_{-\infty}$ on $[\mathbf{p_{new}} - d_{max}, \mathbf{p_{new}} + d_{max}]$
**end for**

# Optimization-based methods



# Optimization-based methods

Second optimization-based criterion used: `optimizer-intersite`

1. Similar to `optimizer-proj`
2. First rank by minimum projected distance
3. Then wiggle ($\alpha = 0.5$ chosen) to maximize minimum intersite distance

# Results

# Summary of methods

Methods (12 total)

- Existing non-sequential methods
  1. `factorial`
  2. `lhd-optimal`
- Existing sequential methods
  1. `lhd-nested`
  2. `voronoi`
  3. `delaunay`
  4. `random`
  5. `halton`
  6. `sobol`
- Novel sequential methods
  1. `mc-intersite-proj`
  2. `mc-intersite-proj-th`
  3. `optimizer-intersite`
  4. `optimizer-proj`

# Test Particulars

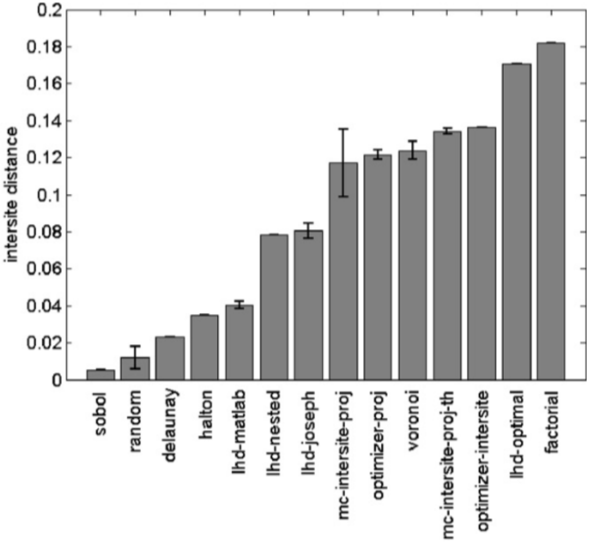
- Methods used to generate 144 points for $d = 2$, 3, and 4
- 15 min max run time
- Each method in each dimension run 30 times to get std dev estimate
- Methods compared on three criteria
  1. Granularity (no. points added per iteration)
  2. Space-filling (intersite distance)
  3. Non-collapsing (projected distance)
- Each novel method has best possible granularity
- Sequential methods expected to perform worse than one-shot methods
  - One-shot methods assume total no. points known beforehand

# Results

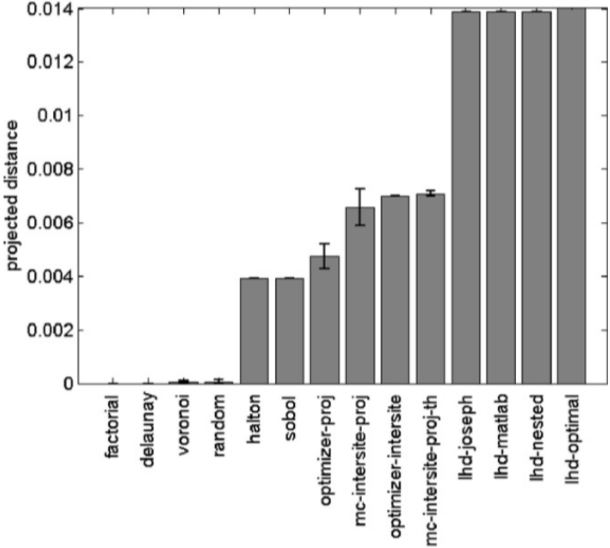
Some important observations

- $d = 2$: Compare `lhd-optimal` to `factorial`
- $d = 2$: Difference between `mc-intersite-proj` and `mc-intersite-proj-th`
- $d = 2, 3, 4$: Compare `optimizer-intersite` to `lhd-optimal`
  - $d = 2$: Performs 21% worse
  - $d = 3$: Performs 16% worse
  - $d = 4$: Performs 8% worse
    - 15 min vs 6 h

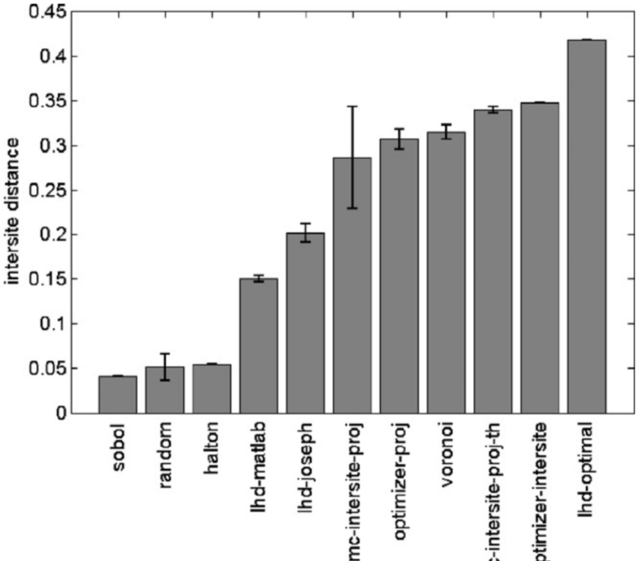
# Results for $d = 2$ (intersite distance)



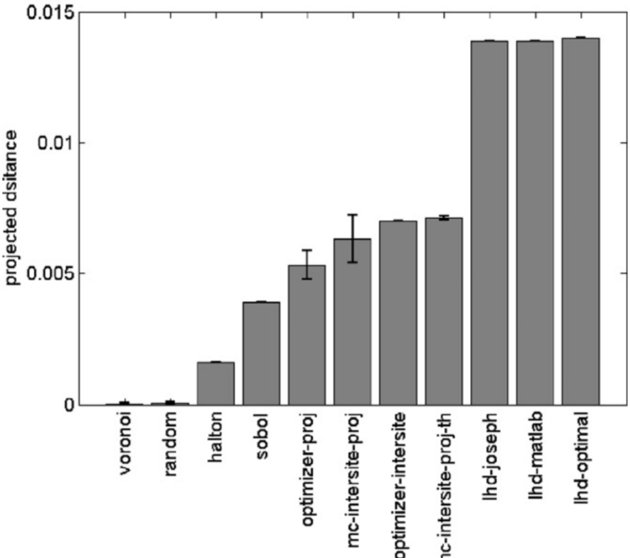
# Results for $d = 2$ (projected distance)



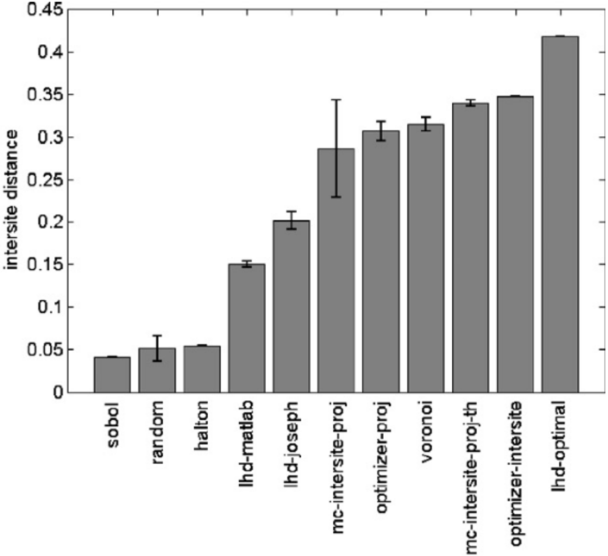
# Results for $d = 3$ (intersite distance)



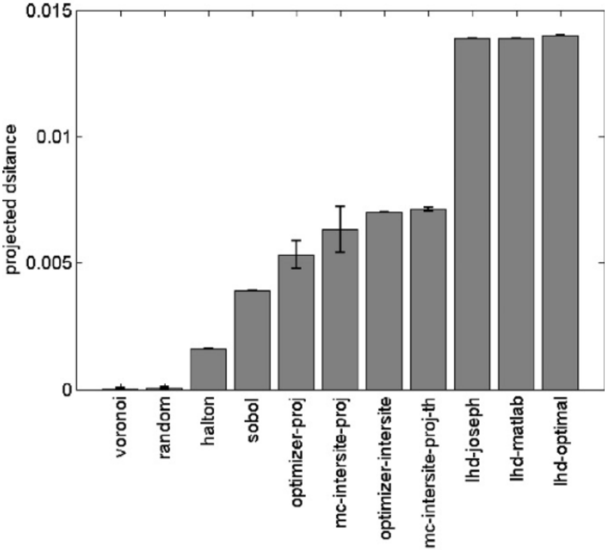
# Results for $d = 3$ (projected distance)



# Results for $d = 4$ (intersite distance)



# Results for $d = 4$ (projected distance)



# Conclusions

# Summary of Results

- New methods perform close to pre-optimized LHD (and much faster)
- Of new methods, best are `optimizer-intersite` and `mc-intersite-proj-th`
  - `optimizer-intersite` possibly unfeasible in higher dimensions
  - `mc-intersite-proj-th` easy to implement, fast, performs well in all dimensions

# References

# References


- K. Crombecq, E. Laermans, T. Dhaene (2011). *Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling.*
- K. Crombecq, I. Couckuyt, D. Gorissen and T. Dhaene (2009). *Space-Filling Sequential Design Strategies for Adaptive Surrogate Modelling.*

# Thank you!

Questions? Comments? Critiques? (I have some critiques for the paper)