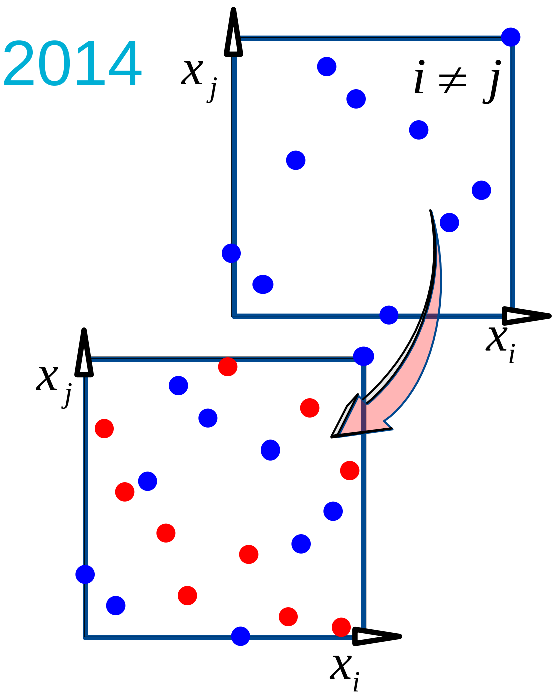
A Flexible Strategy for Augmenting Design Points For Computer Experiments

7th Dresdner Probabilistik Workshop, 9th Oct 2014



Peter M. Flassig*, Ron A. Bates**

*Rolls-Royce Deutschland Ltd & Co KG

** Rolls-Royce plc

Trusted to deliver excellence

13/10/2014

Rolls-Royce

Starting Point – An industrial Challenge

Let $f : R^d \longrightarrow R$ be a costly to evaluate physical system to be modelled in short time.

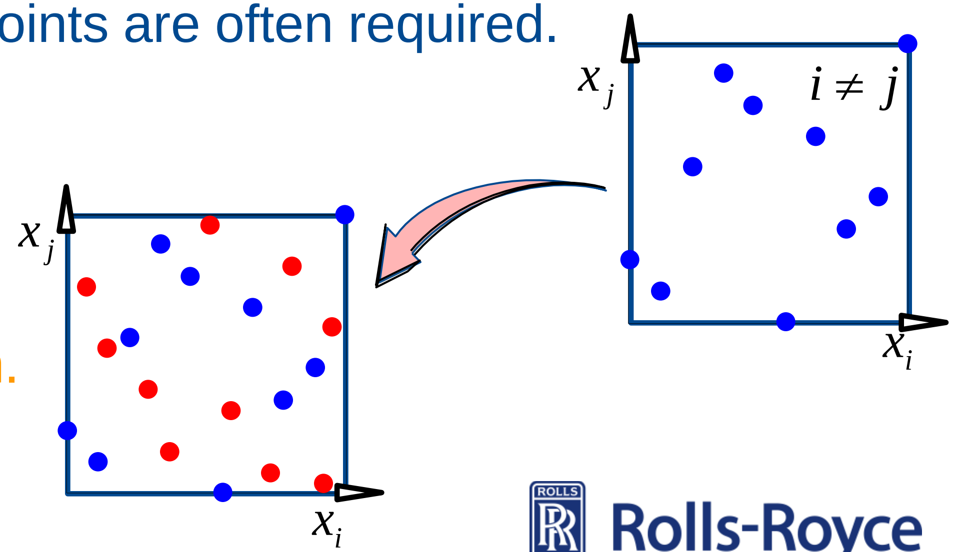
Common Solution: Run a DoE.

Goal: Invest as few as possible design evaluations N to obtain a sufficient understanding, i.e. $\min_N N \text{ s.t. } |f(\mathbf{x}) - \hat{f}(\mathbf{x})| \leq \varepsilon$

emulated understanding (→ $\hat{f}(\mathbf{x})$) *error* (→ ε)

Problem: In practice, additional points are often required.

Construct an *optimal* augmented design plan.

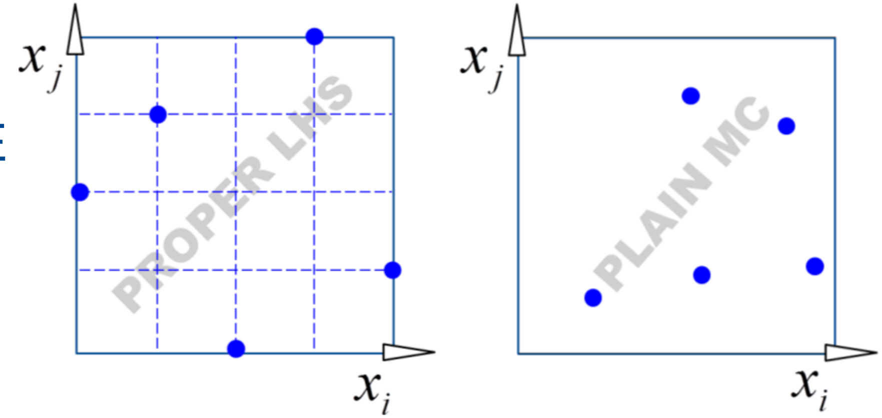


Rolls-Royce

Now, from an industrial perspective an augmentation algorithm needs to

(i) be UNIVERSAL, i.e.

- independent from type of initial DoE



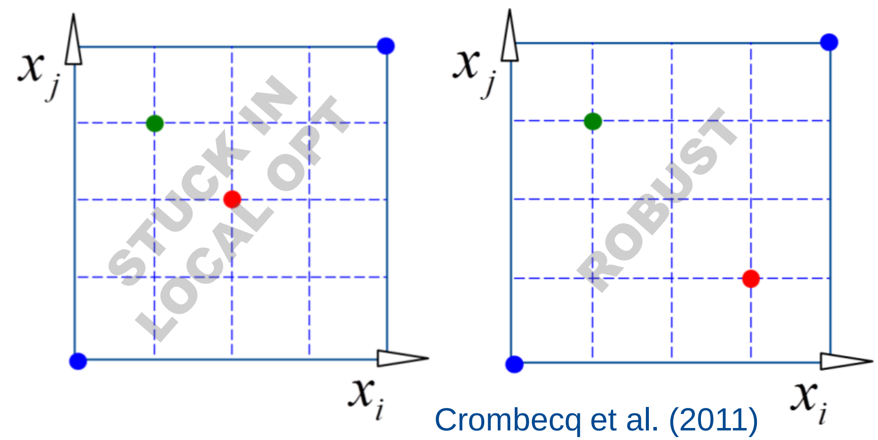
- independent from selected augmentation sequence, i.e. number of levels l and batch size m

Example with $l = 2$ and $m = 1$:

$N_0 = 2$ (initial points)

$N_1 = 3$ (first augmentation)

$N_2 = 4$ (second augmentation)



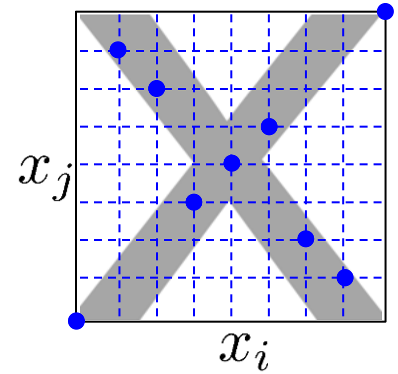
Crombecq et al. (2011)

- not restricted to a specific range of design space dimension d or final sample plan size $N = N_0 + \sum_{i=1}^{l} m_i$

Rolls-Royce

(ii) improve SPACE FILLING, i.e.

- focus on exploration because no model available at the beginning
- evenly spread of design points in design space (avoid X-problem)
- related to projection and orthogonal properties



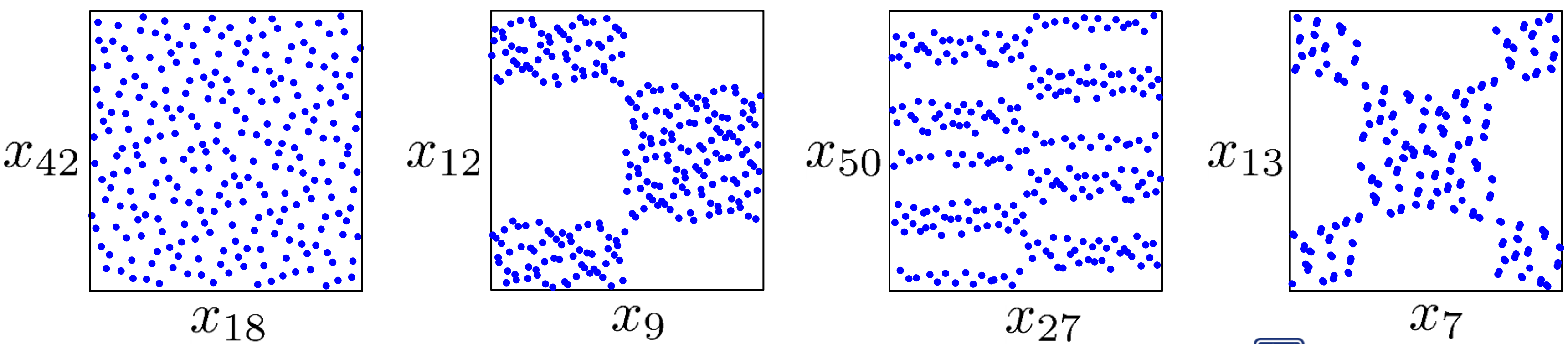
Hernandez et al. (2012)

(iii) be HIHGLY EFFICIENT

- even for $d > 50$
- w.r.t. computational time t

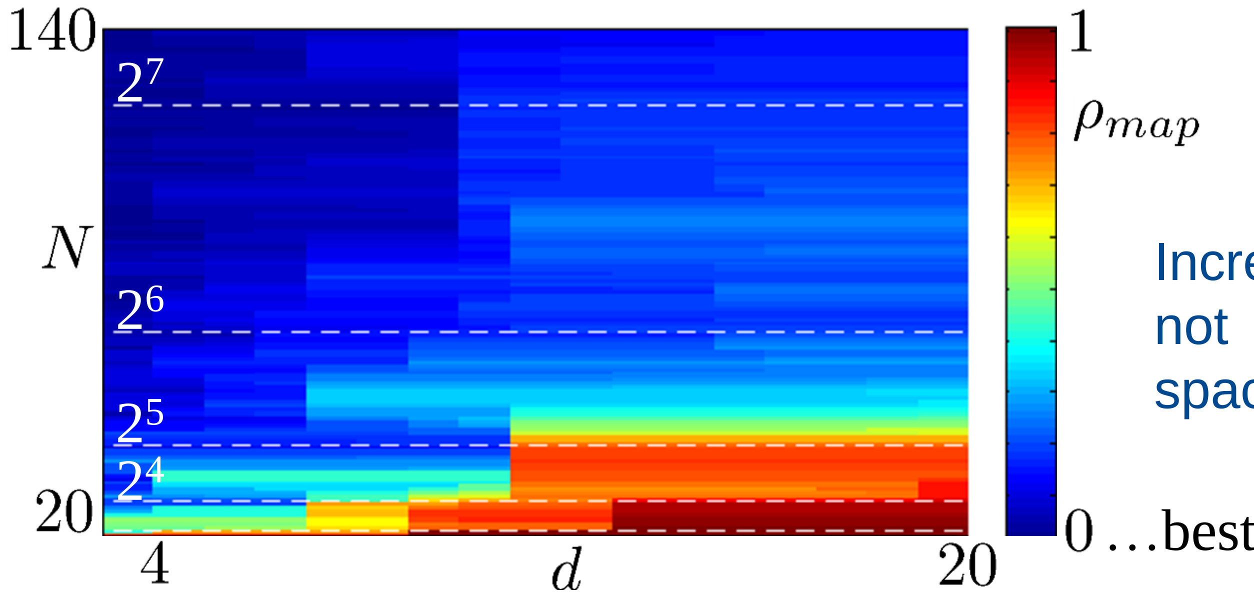
Why e.g. Sobol could fail?

Do some random illustrations for $d = 50 \quad N = 256$



Rolls-Royce

Evaluate maximum absolute pairwise (map) correlation coefficient defined as $\rho_{map} = \max_{1 \leq (i,j) \leq d, i \neq j} |\rho_{ij}^P|$ for different d and N



Increase of points does not lead to improved space filling a priori.

Can we do better?

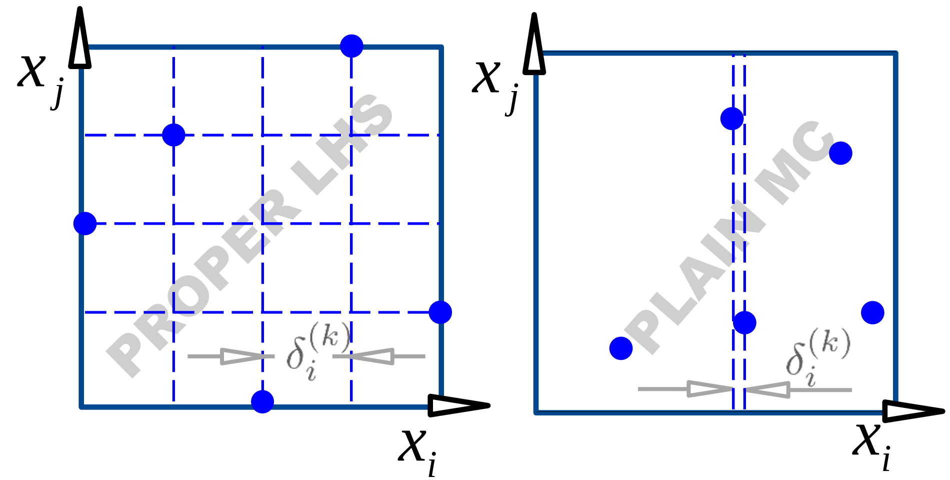
Outline

- Augmentation of (i) an integer grid and (ii) a general grid
- *Optimal* space filling by brute force approach
- Comparative examples

Rolls-Royce

Augmentation on Integer or General Grid?

Let $\boldsymbol{X}^{(k)}$ be the k-th sample of a DoE, $k=1(1)N$ and $\boldsymbol{x}=[x_1,\ldots,x_d]^{\mathrm{T}}$ $[0,1]^d$, to be augmented by m points.



Classify type of input design plan by strata widths defined as

$$\delta_i^{(k)} = x_i^{(k+1:N)} - x_i^{(k:N)} \quad \text{where} \quad x_i^{(1:N)} < x_i^{(2:N)} < \ldots < x_i^{(N:N)}$$

$$\text{with} \quad k = 1\,(1)\,N-1 \quad \text{and} \quad i = 1\,(1)\,d$$

Compute decision criterion $\delta = \delta_i^{(k)} - \left\lfloor \delta_i^{(k)} / \min_k \delta_i^{(k)} \right\rfloor \min_k \delta_i^{(k)}$

Do integer grid augmentation if $\delta \leq 10^{-6}$ and general grid augmentation if $\delta > 10^{-6}$.

Rolls-Royce

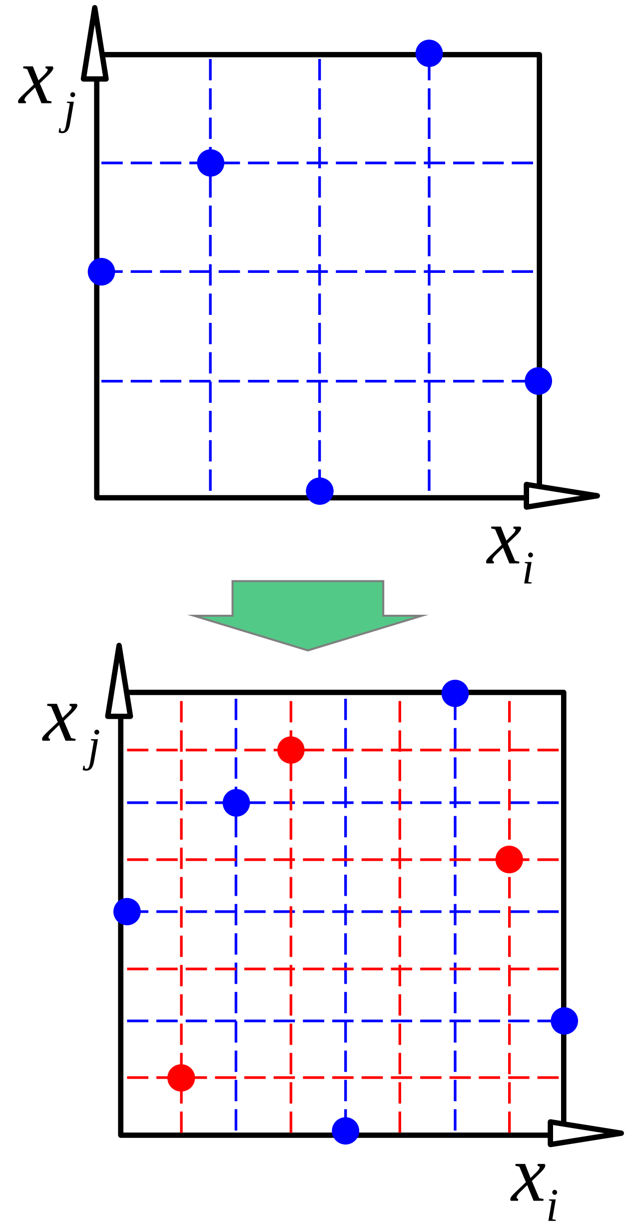
Integer Grid Augmentation

Check extensibility of the grid by

$$\left\lfloor \max_k \frac{1}{\delta_i^{(k)}} + \frac{1}{2} \right\rfloor + 1 < N + m$$

If necessary refine the grid. Do identification and indexing of all possible positions for new points.

Generate sets of d random perturbations $\{1,2,\ldots,1/(\delta^{\min}-1)-N\}$ repeatedly & select set which provides best space-filling criterion.



Rolls-Royce

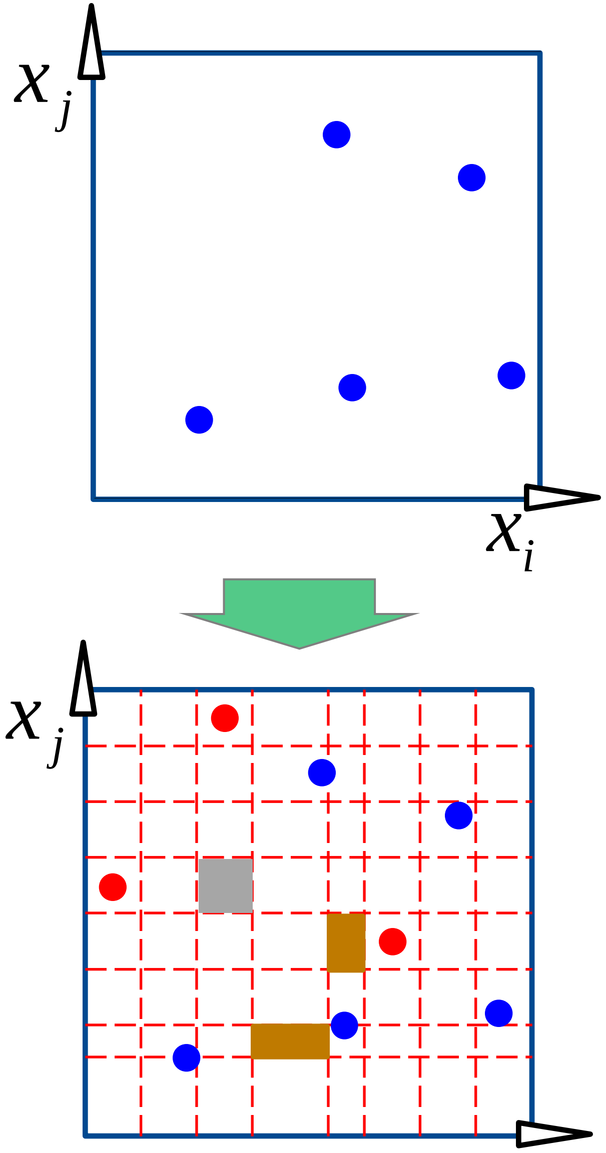
General Grid Augmentation

Divide dimensions in $N + m$ equally distributed strata.

Identification of current strata positions for all designs.

From **HYPERCUBES** to **HYPER-CUBOIDS**: Adjust strata bounds till each stratum is used by maximal one design point only.

Generate sets of d random perturbations $\{1,2,\ldots,m\}$ repeatedly and select set according to best space-filling criterion to fill empty positions.



Optimal Space-filling by Brute Force Approach

How to access quality of a design plan?

- maximum absolute pairwise (map) correlation coefficient (to be minimal)

$$\rho_{map} = \max_{1 \le (i,j) \le d, i \ne j} |\rho_{ij}^P|$$

- modified L_2 discrepancy (to be minimal)

$$ML_2 = \left(\frac{4}{3}\right)^d - \frac{2^{1-d}}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \left(3 - x_{ij}^2\right) + \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \prod_{l=1}^{d} \left(2 - \max\left\{x_{ij}, x_{jl}\right\}\right)$$

Crombecq et al. (2011),
Joseph and Hung (2008)

Rolls-Royce

How to optimize?

- objective function is nonlinear, not differentiable everywhere
- design parameters are natural numbers, i.e. integers
- lattice construct has $N!^d$ possible Lhs
- for some combinations of N and d *complex* algorithms exist

Let use a brute force approach.

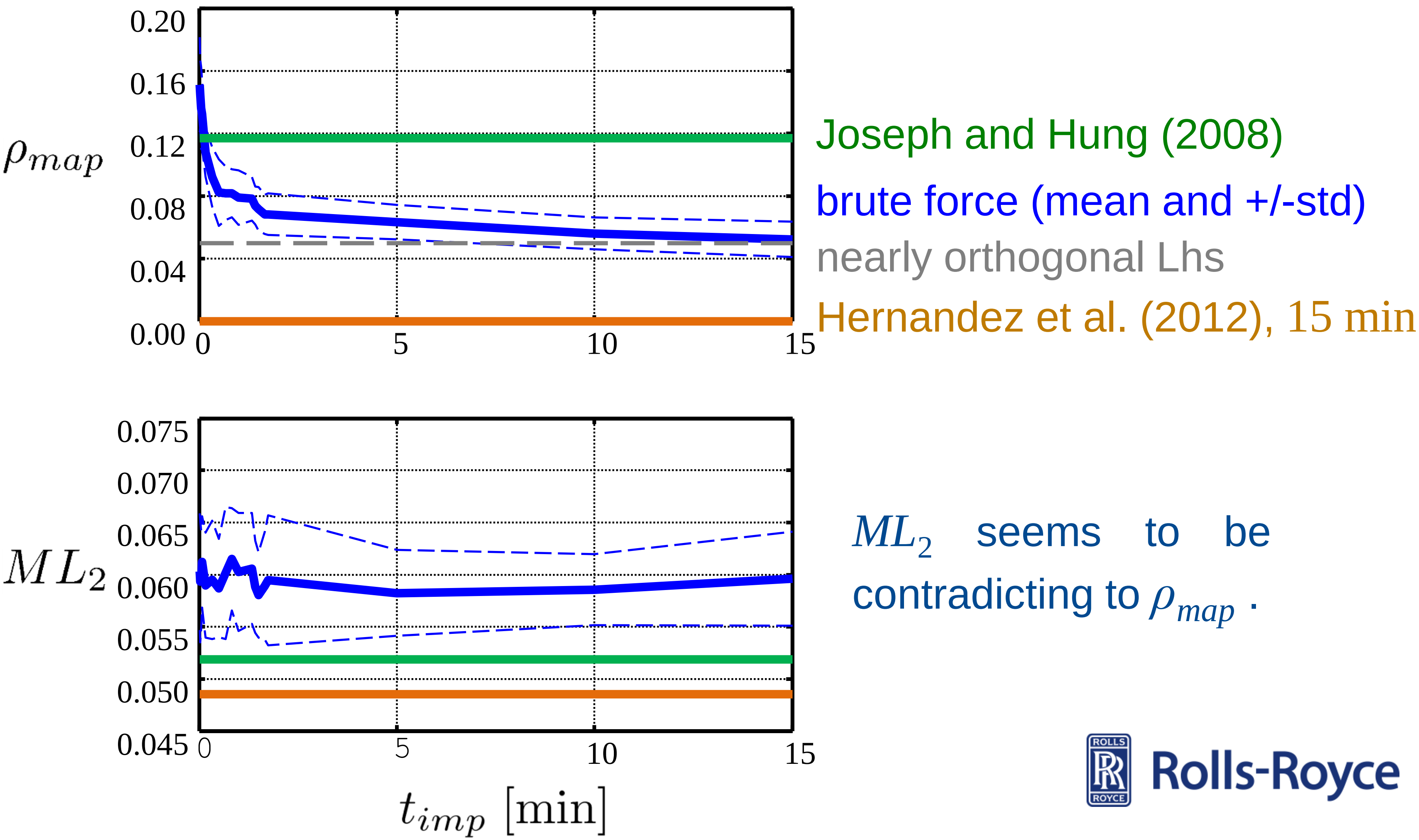
Generate repeatedly permutations for a certain amount of time and select best set of permutations which provides best quality.

Is this compatible?

Comparison with results published by Hernandez et al. (2012) and Joseph and Hung (2008) for $d = 4$ and $N = 9$ obtained by complex algorithms.

1st Scenario

- brute force by minimizing of ρ_{map} only
- vary improving time t_{imp} from one second to 15 minutes
- do 30 repetitions for each time do enable statistical statements

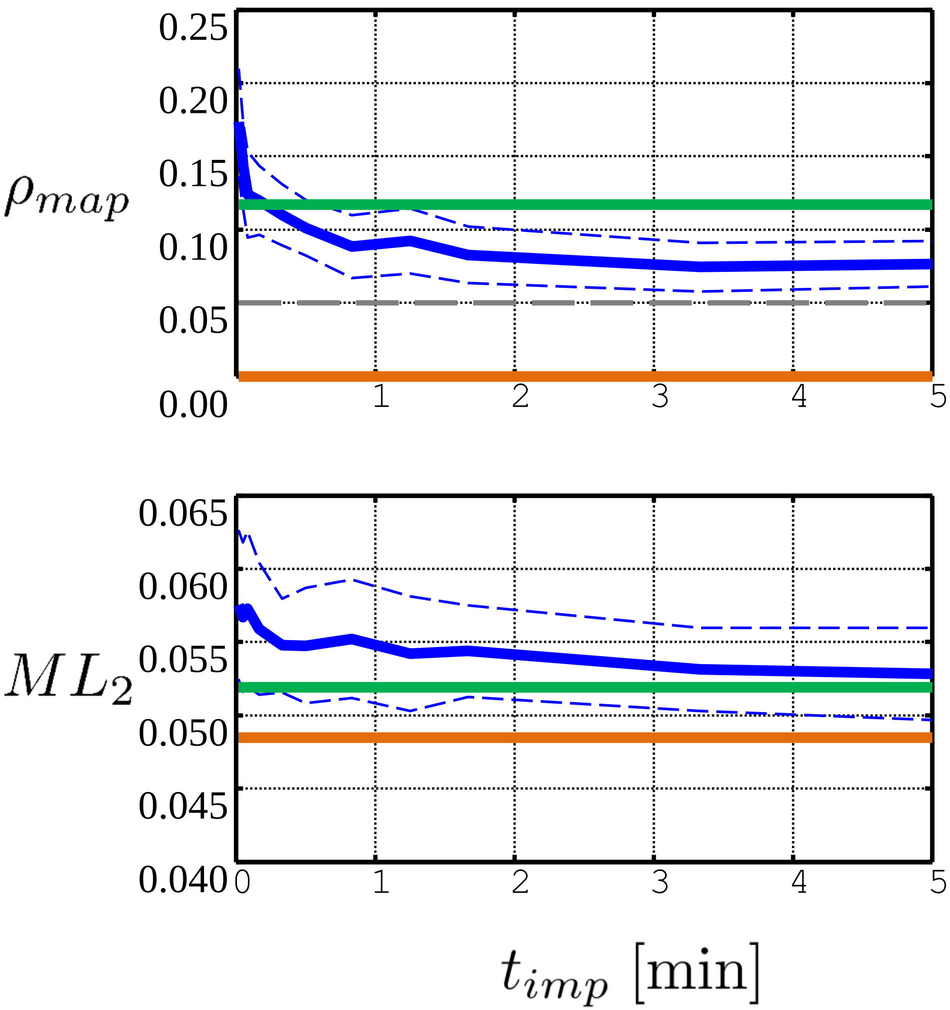


ML_2 seems to be contradicting to ρ_{map} .

Rolls-Royce

2nd Scenario

- brute force by minimizing of $0.2\rho_{map} + 0.8ML_2$
- vary improving time t_{imp} from one second to five minutes
- do 30 repetitions for each time do enable statistical statements



Joseph and Hung (2008)

brute force (mean and +/-std)

nearly orthogonal Lhs

Hernandez et al. (2012), 15 min

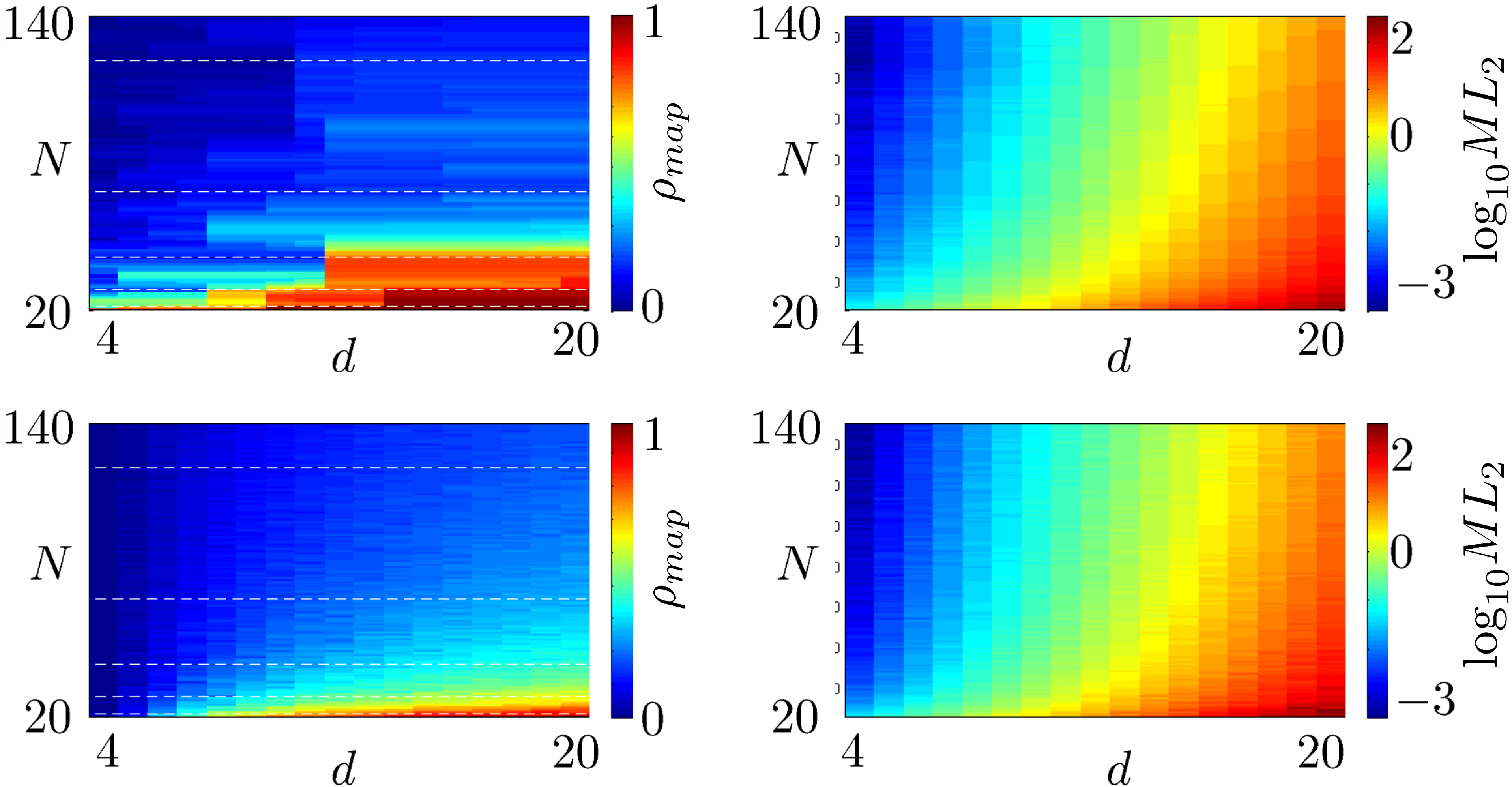
Proper weighting coefficients lead to a desired trade off.

Rolls-Royce

A First Comparative Example

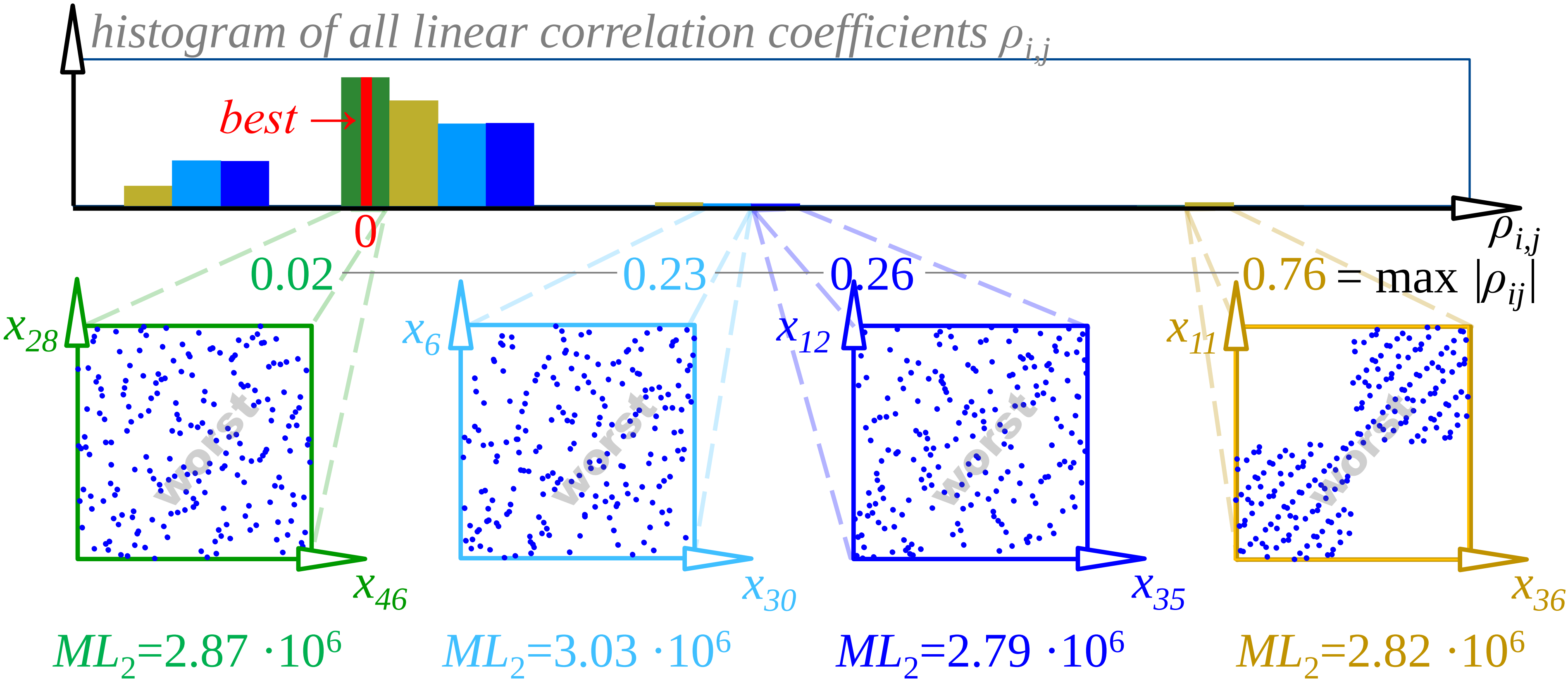
Compare a SOBOL (top) generator with brute force improved Lhs (bottom)

$d \in [3, 20] \quad N \in [5, 150] \quad t_{imp} = 10\, sec \quad 17 \times 145\, \text{grid}$



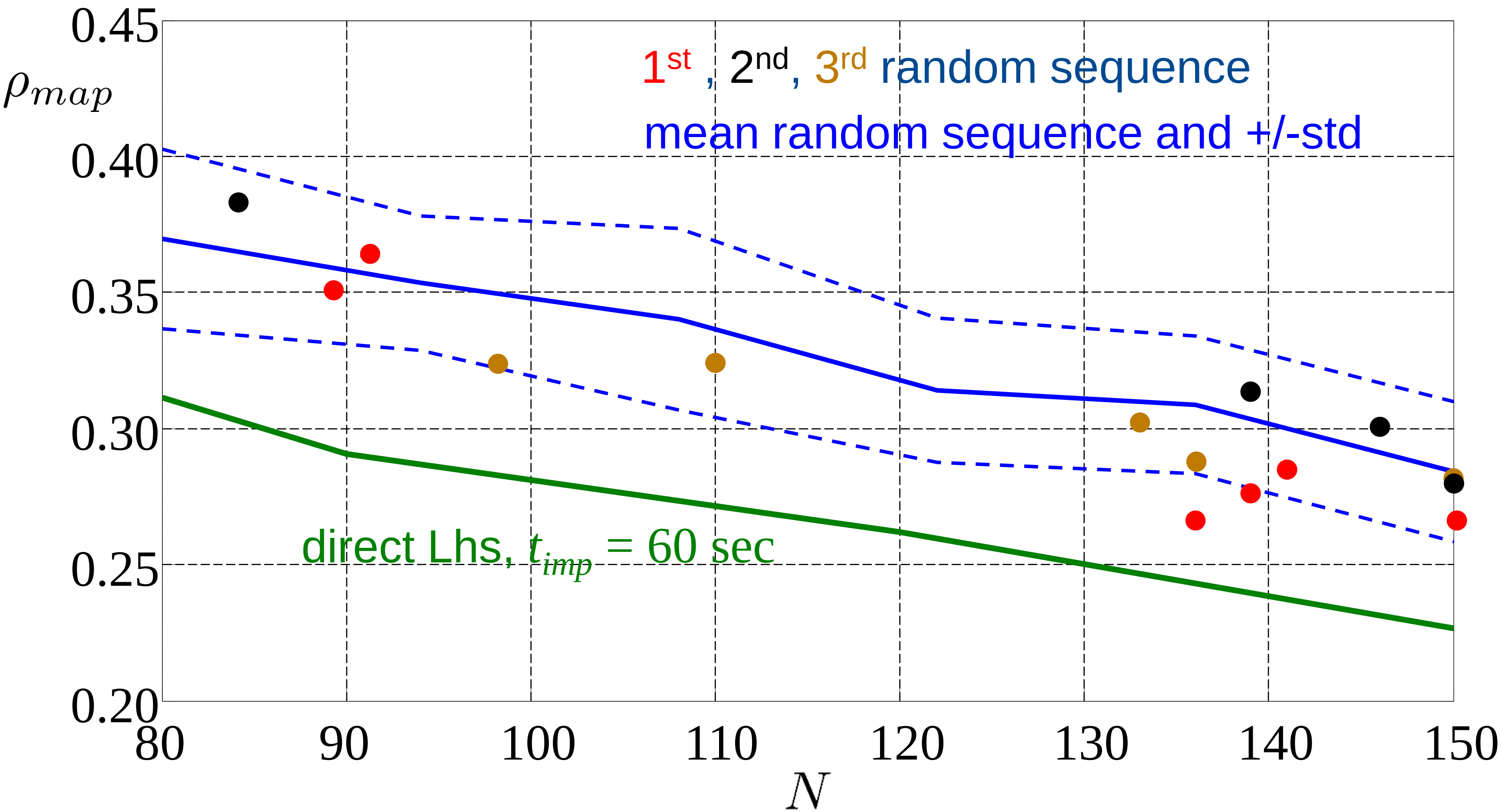
A Second Comparative Example

Let d = 50, N_0 = 60, N_i = $N_{i\text{-}1}$ + m_i with $\mathbf{m}$ = $[40,50,50]^T$ be an augmentation sequence applied to an initial optimal Lhs (6.6min) and an initial plain MC (4.1min). Compare augmented plans with one optimal Lhs design plan (20min) and one SOBOL design plan (0.1min) with N=200.



A Third Comparative Example

Do repeatedly random augmentation sequences (100 times) of an initial Lhs design plan for $d = 50$, $N_0 = 64$, $N_l = 150$ and $t_{imp} = 10$ sec, i.e. generate randomly number of levels l and number of points m_i to be added at each level.



Summary and Outlook

A universal algorithm for sequentially augmenting computer experiments is presented. Preliminary results show potentials related to space-filling even for design space dimensions of $d = 50$.

Test proposed method against other quasi-random low-discrepancy sequences.

Implement a capability to add a factor/design parameter instead of augmenting number of samples.

References

K. Crombecq, E. Laermans, T. Dhaene. 2011. Efficient space-filling and non-collapsing design strategies for simulation-based modeling. *European Journal of Operational Research 214*, 683-696.

ALEJANDRO S. HERNANDEZ, THOMAS W. LUCAS, and MATTHEW CARLYLE. 2012. Constructing Nearly Orthogonal Latin Hypercubes for Any Nonsaturated Run-Variable Combination. *ACM Transactions on Modeling and Computer Simulation 22*, 4, 20:1-20:17.

V. Roshan Joseph and Ying Hung. 2008. ORTHOGONAL-MAXIMIN LATIN HYPERCUBE DESIGNS. *Statistica Sinica 18*, 171-186.

Rolls-Royce

Back Up Slides

Average number of generated Lhs design plans for 1st Scenario.