Non-Collapsing Space-Filling Designs for Bounded Regions

Angela Dean

The Ohio State University

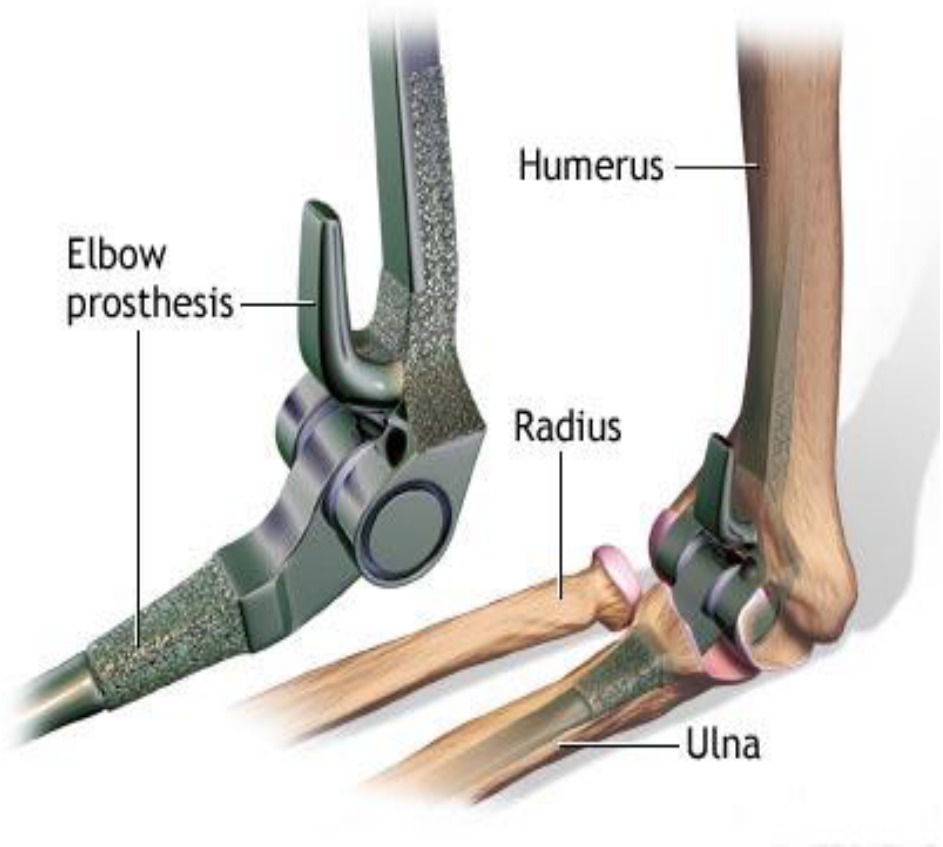
September, 2011

Isaac Newton Institute, Cambridge

Joint Work with Danel Draguljić and Thomas Santner

Motivating Example 1: Total elbow arthroplasty

Hayeck (2009) modeled compressive strains in the bone as a function of humeral implant position for a total elbow prosthesis



Motivating Example 1: Total elbow arthroplasty

- Used $p = 4$ variables to specify implant position
  - x_1 - tip displacement (mm).
  - x_2 - rotation of the implant axis about the lateral axis (°)
  - x_3 - rotation of the implant axis about the anterior axis (°) .
  - x_4 - rotation about the implant axis (°).

Motivating Example 1: Total elbow arthroplasty

- Used $p = 4$ variables to specify implant position
  - x_1 - tip displacement (mm).
  - x_2 - rotation of the implant axis about the lateral axis ($^\circ$)
  - x_3 - rotation of the implant axis about the anterior axis ($^\circ$) .
  - x_4 - rotation about the implant axis ($^\circ$).
- Constraints: max tip displacement is 10 *mm*; rotation of implant axis is 10° about lateral axis at tip and 4° about anterior axis at tip; rotation about the implant axis is $\pm 15^\circ$:

Motivating Example 1: Total elbow arthroplasty

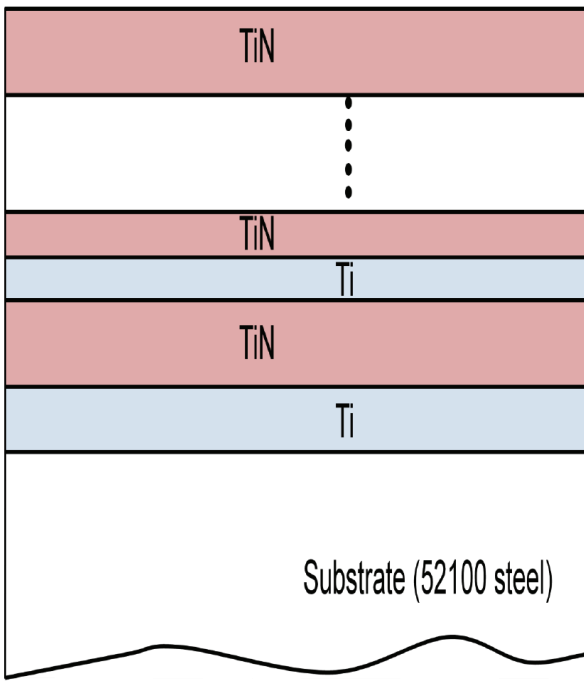
- Used $p = 4$ variables to specify implant position
  - x_1 - tip displacement (mm).
  - x_2 - rotation of the implant axis about the lateral axis (°)
  - x_3 - rotation of the implant axis about the anterior axis (°) .
  - x_4 - rotation about the implant axis (°).
- Constraints: max tip displacement is 10 *mm*; rotation of implant axis is 10° about lateral axis at tip and 4° about anterior axis at tip; rotation about the implant axis is $\pm 15°$:

$$\begin{array}{rcl} 0 \leq & x_1 & \leq 10 \\ & 5x_2 + 2x_3 & \leq 10 \\ & -5x_2 + 2x_3 & \leq 10 \\ -10 \leq & 5x_2 + 2x_3 & \\ -10 \leq & -5x_2 + 2x_3 & \\ -15 \leq & x_4 & \leq 15 \end{array}$$

Motivating Example 2: Multilayer Tool Coating

Nekkanty (2009) investigated the relationship between peak normal stress (cohesive cracks) and peak shear stress (delamination) as a function of number of coats and thickness. A coating design is defined by $(p, x_1, \ldots, x_p)$ where

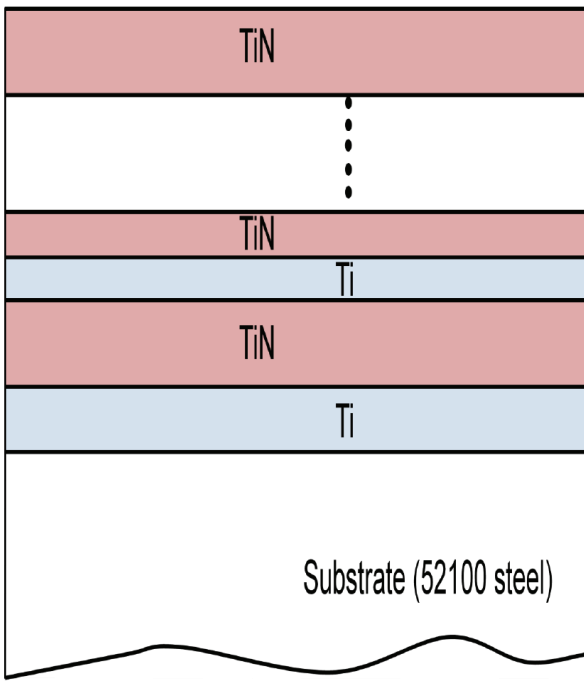
$$p \in \{2, 4, 6, 8\} \text{ and } 2\ \mu m \leq \sum_{i=1}^{p} x_i \leq 6\ \mu m \text{ and } .25\ \mu m \leq x_i$$

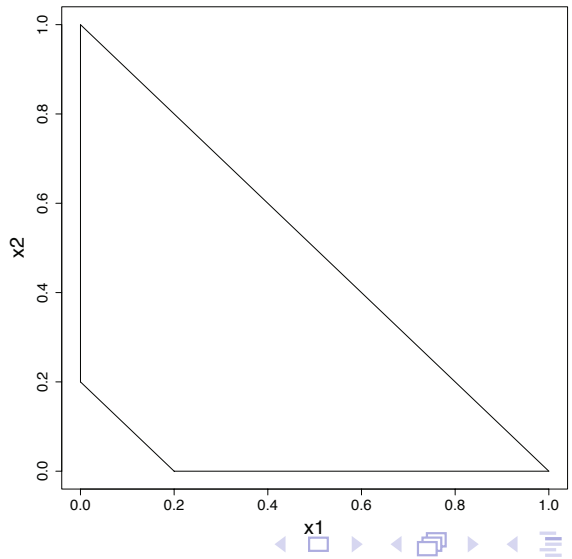


Motivating Example 2: Multilayer Tool Coating

Nekkanty (2009) investigated the relationship between peak normal stress (cohesive cracks) and peak shear stress (delamination) as a function of number of coats and thickness. A coating design is defined by $(p, x_1, \ldots, x_p)$ where

$$p \in \{2, 4, 6, 8\} \text{ and } 2\ \mu m \leq \sum_{i=1}^{p} x_i \leq 6\ \mu m \text{ and } .25\ \mu m \leq x_i$$





Design required

In each of these examples:

- a design was required at which to run the computer code that implemented a mathematical model describing the physical process (i.e. a computer experiment)

Design required

In each of these examples:

- a design was required at which to run the computer code that implemented a mathematical model describing the physical process (i.e. a computer experiment)
- *Common design approach*: for fitting flexible process-based predictors (e.g. GaSP model), select design points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ to be as space-filling as possible within the input region

Design required

In each of these examples:

- a design was required at which to run the computer code that implemented a mathematical model describing the physical process (i.e. a computer experiment)
- *Common design approach*: for fitting flexible process-based predictors (e.g. GaSP model), select design points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ to be as space-filling as possible within the input region
- The literature suggests numerous criteria that might be used to achieve space-filling designs

Design required

In each of these examples:

- a design was required at which to run the computer code that implemented a mathematical model describing the physical process (i.e. a computer experiment)
- *Common design approach*: for fitting flexible process-based predictors (e.g. GaSP model), select design points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ to be as space-filling as possible within the input region
- The literature suggests numerous criteria that might be used to achieve space-filling designs
- This talk uses a combination of the Maximin and ARD criteria

Space-filling Criteria

- Maximin - A design $\boldsymbol{X}_{Mm}$ is maximin (Mm) provided it maximizes the minimum interpoint distance (mIPD) among all designs in the class, $\mathcal{D}$, of n-point designs

$$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2 \in \boldsymbol{X}_{Mm}} \rho_z(\boldsymbol{x}_1, \boldsymbol{x}_2) = \max_{\boldsymbol{X} \in \mathcal{D}} \left[\min_{\boldsymbol{x}_1,\boldsymbol{x}_2 \in \boldsymbol{X}} \rho_z(\boldsymbol{x}_1, \boldsymbol{x}_2) \right]$$

where

$$\rho_z(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(\sum_{\ell=1}^{p} |x_{i\ell} - x_{j\ell}|^z \right)^{1/z}$$

Space-filling Criteria

- Average Reciprocal Distance with respect to set of dimensions $\boldsymbol{J} \subset \{1, \ldots, p\}$

  A design $\boldsymbol{X}_{ARD}$ is ARD optimal wrt $\boldsymbol{J}$ (ARD/$\boldsymbol{J}$) provided it minimizes

$$av_z(\boldsymbol{X}) = \frac{1}{\binom{n}{2} \sum_{j \in \boldsymbol{J}} \binom{p}{j}} \sum_{j \in \boldsymbol{J}} \sum_{k=1}^{\binom{p}{j}} \sum_{\boldsymbol{x}_h^\star, \boldsymbol{x}_i^\star \in \boldsymbol{X}_{kj}} \left[\frac{j^{1/z}}{\rho_z(\boldsymbol{x}_h^\star, \boldsymbol{x}_i^\star)} \right]$$

  where the set $\boldsymbol{J}$ specifies the set of sub-dimensions over which the reciprocal average is computed and $\boldsymbol{x}_h^\star, \boldsymbol{x}_i^\star, \boldsymbol{X}_{kj}$ are the projections of $\boldsymbol{x}_h, \boldsymbol{x}_i, \mathbf{X}$ onto the kth subspace of dimension j.

Updating Formula

To save computing time, we use an updating formula when adding a new point **w** to the design:

$$av_z\left(\begin{bmatrix}\mathbf{X}\\ \mathbf{w}\end{bmatrix}\right) = \frac{n-2}{n} av_z(\mathbf{X}) + \Delta$$

where

$$\Delta = \frac{1}{\binom{n}{2}\sum_{j\in \boldsymbol{J}}\binom{p}{j}} \sum_{j\in \boldsymbol{J}}\sum_{k=1}^{\binom{p}{j}}\left[\sum_{i=1}^{n-1}\frac{j^{1/z}}{\rho_z(\mathbf{x}_i^\star, \mathbf{w})}\right]$$

so that only distances involving the new point **w** need be computed

Space-filling Criteria

This talk uses a convex combination of mIPD and ARD; that is, given $0 \leq \alpha \leq 1$, minimize

$$\alpha R_{mIPD} + (1 - \alpha) R_{ARD}$$

where R represents a rank for a particular design and criterion among all designs under consideration.

In addition to space filling:

- In many applications, large numbers of inputs are considered initially in a screening stage; e.g.
  - engineering design inputs
  - field/operating inputs,
  - model/calibration inputs

In addition to space filling:

- In many applications, large numbers of inputs are considered initially in a screening stage; e.g.
  - engineering design inputs
  - field/operating inputs,
  - model/calibration inputs
- Not all expected to have major impact on the response

In addition to space filling:

- In many applications, large numbers of inputs are considered initially in a screening stage; e.g.
  - engineering design inputs
  - field/operating inputs,
  - model/calibration inputs
- Not all expected to have major impact on the response
- For deterministic computer runs, replication is unnecessary

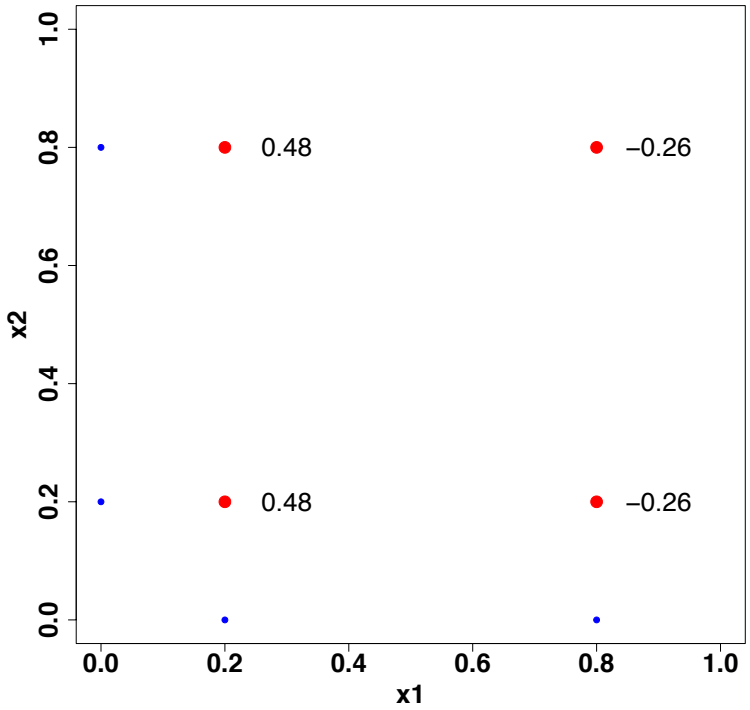
In addition to space filling:

- In many applications, large numbers of inputs are considered initially in a screening stage; e.g.
  - engineering design inputs
  - field/operating inputs,
  - model/calibration inputs
- Not all expected to have major impact on the response
- For deterministic computer runs, replication is unnecessary
- Therefore, in addition to choosing $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ to be space-filling, we want designs which do not have replication when projected on subsets of factors

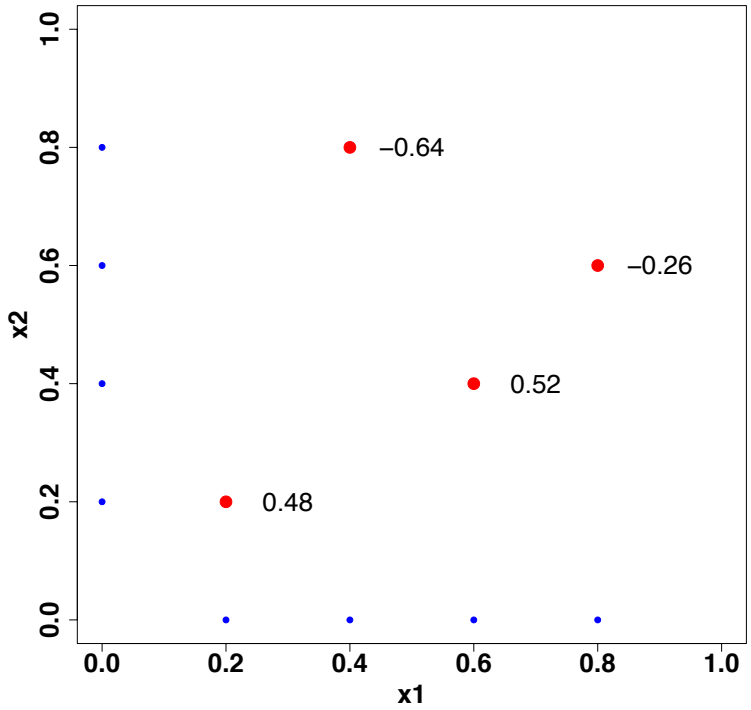
In addition to space filling:

- In many applications, large numbers of inputs are considered initially in a screening stage; e.g.
  - engineering design inputs
  - field/operating inputs,
  - model/calibration inputs
- Not all expected to have major impact on the response
- For deterministic computer runs, replication is unnecessary
- Therefore, in addition to choosing $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ to be space-filling, we want designs which do not have replication when projected on subsets of factors
- We call such designs non-collapsing

Computer Experiment Design

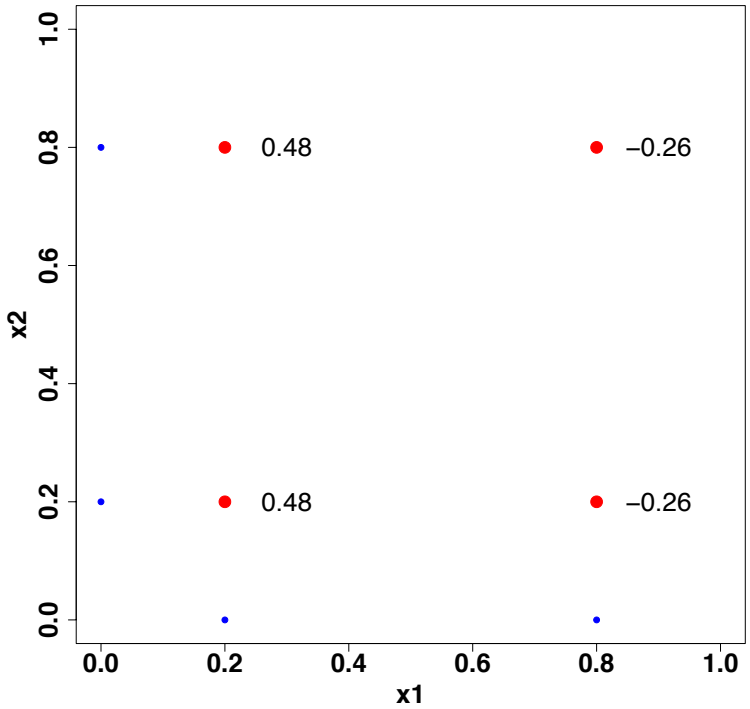


(a) Collapsing Design

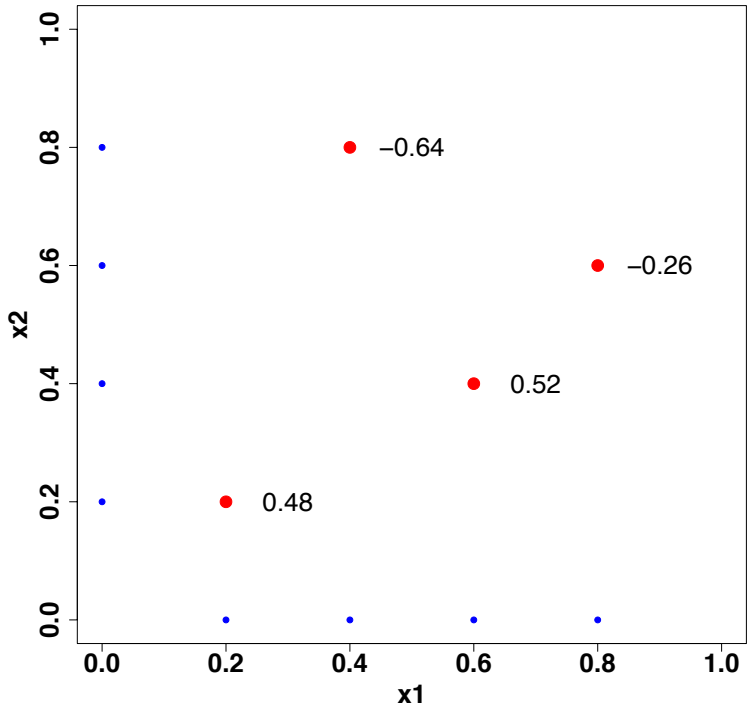


(b) Non-collapsing Design

Computer Experiment Design



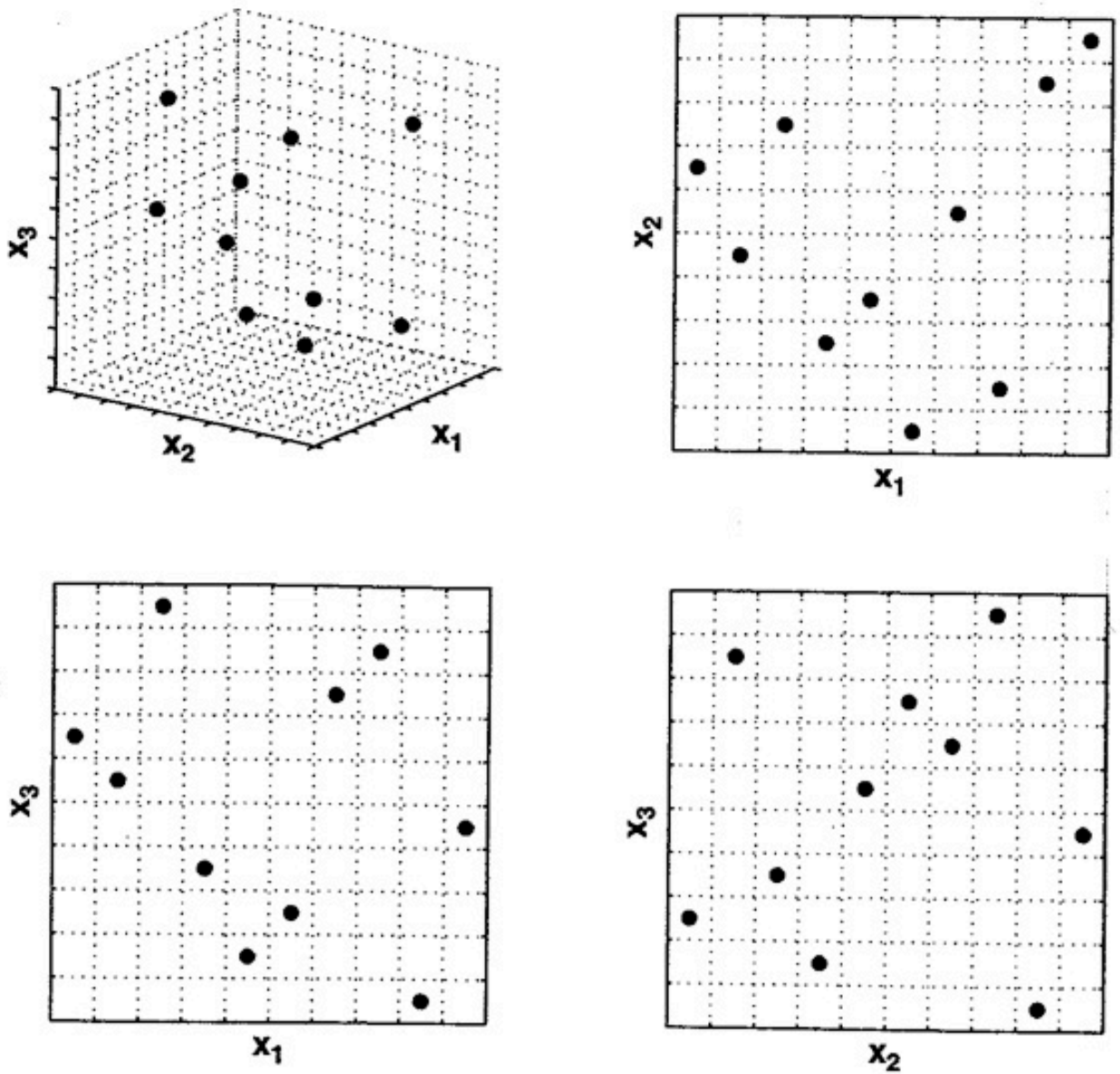
(c) Collapsing Design



(d) Non-collapsing Design

- Latin Hypercube Designs are examples of non-collapsing designs over $[0, 1]^p$

LHD example (Forrester, Sobester, Keane, 2008)



Constrained Experimental Region

- But in our examples, we cannot use a LHD since the input region is a bounded set of the form

$$\boldsymbol{Ax} \leq \boldsymbol{b} \text{ and } \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U}$$

Constrained Experimental Region

- But in our examples, we cannot use a LHD since the input region is a bounded set of the form

$$\boldsymbol{Ax} \leq \boldsymbol{b} \text{ and } \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U}$$

- Construction of space-filling and non-collapsing designs is difficult because
  - columns of the design matrix $\boldsymbol{X}$ cannot be chosen independently
  - high dimensionality precludes complete search over a fine grid

Approaches in the Literature for Constrained Regions

We found three approaches in the literature

Approaches in the Literature for Constrained Regions

We found three approaches in the literature

- Trossett (1999)
  - Formuates selection of Mm design as NLP problem
  - Designs need not be non-collapsing

Approaches in the Literature for Constrained Regions

We found three approaches in the literature

- Trossett (1999)
  - Formuates selection of Mm design as NLP problem
  - Designs need not be non-collapsing
- Stinstra et. al (2003)
  - Also formulates selection of Mm design as a NLP problem, but sequentially
  - Designs need not be non-collapsing
  - Requires access to specialized software (NLP solver CONOPT).

Approaches in the Literature for Constrained Regions

We found three approaches in the literature

- Trossett (1999)
  - Formuates selection of Mm design as NLP problem
  - Designs need not be non-collapsing
- Stinstra et. al (2003)
  - Also formulates selection of Mm design as a NLP problem, but sequentially
  - Designs need not be non-collapsing
  - Requires access to specialized software (NLP solver CONOPT).
- Chuang and Hung (2010)
  - Applies (Nearly) Uniform design criterion.

Approaches in the Literature for Constrained Regions

We found three approaches in the literature

- Trossett (1999)
  - Formuates selection of Mm design as NLP problem
  - Designs need not be non-collapsing
- Stinstra et. al (2003)
  - Also formulates selection of Mm design as a NLP problem, but sequentially
  - Designs need not be non-collapsing
  - Requires access to specialized software (NLP solver CONOPT).
- Chuang and Hung (2010)
  - Applies (Nearly) Uniform design criterion.
- Our proposed algorithm CoNcaD

Our proposed algorithm: CoNcaD

CoNcaD Algorithm: **Co**nstrained **N**on- **c**oll **a**psing **D**esign

Our proposed algorithm: CoNcaD

CoNcaD Algorithm: **Co**nstrained **N**on- **c**oll **a**psing **D**esign

- CoNcaD builds an $n \times p$ design matrix $\boldsymbol{X}$ column-wise, accounting for input dependency, i.e.
  - choose the values for input x_1,
  - given the x_1 values and constraints, choose x_2 values,
  - given the x_1, x_2 values and constraints, choose x_3 values,
  - etc.

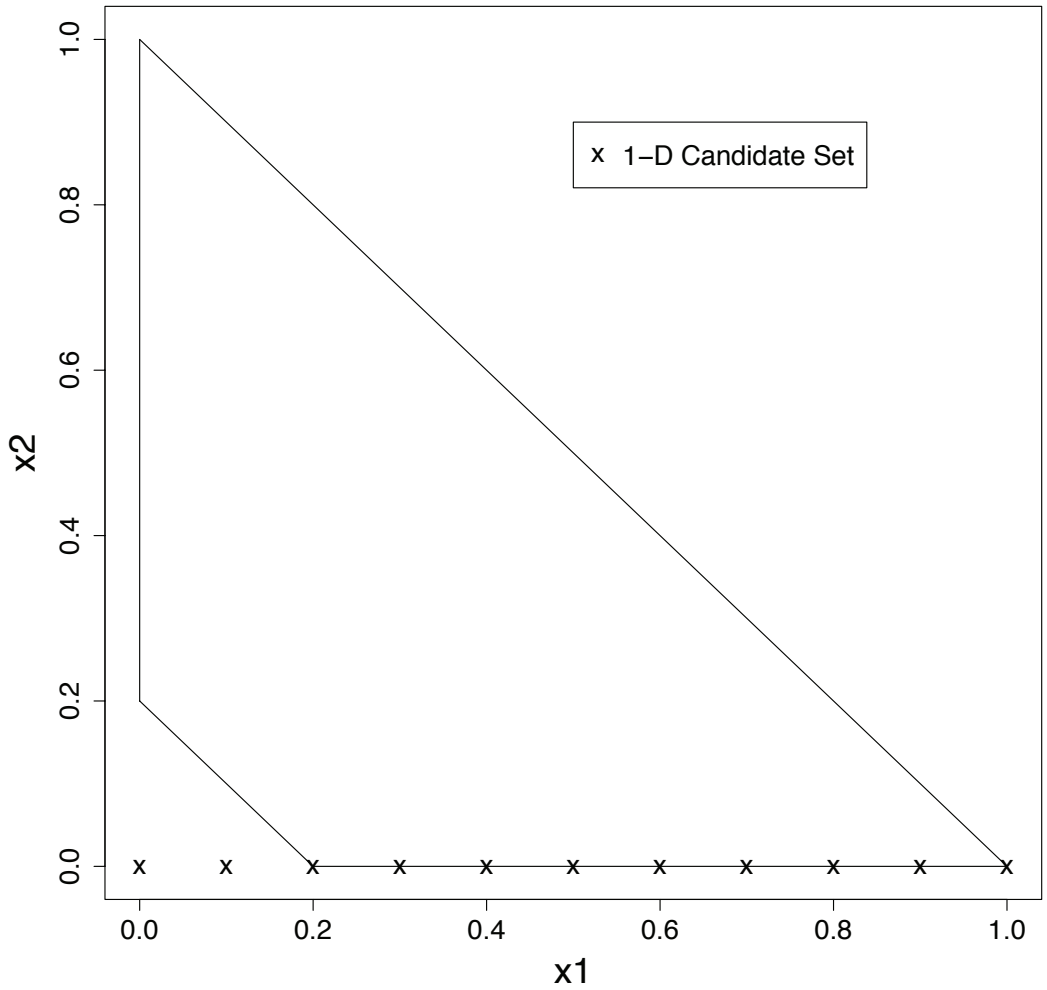
Our proposed algorithm: CoNcaD

CoNcaD Algorithm: **Co**nstrained **N**on- **c**oll **a**psing **D**esign

- CoNcaD builds an $n \times p$ design matrix $\boldsymbol{X}$ column-wise, accounting for input dependency, i.e.
  - choose the values for input x_1,
  - given the x_1 values and constraints, choose x_2 values,
  - given the x_1, x_2 values and constraints, choose x_3 values,
  - etc.
- Features of the algorithm
  - Design is built from a grid which adapts dynamically to enforce non-collapsing
  - Points are added sequentially by optimizing the chosen criterion to enforce space-filling

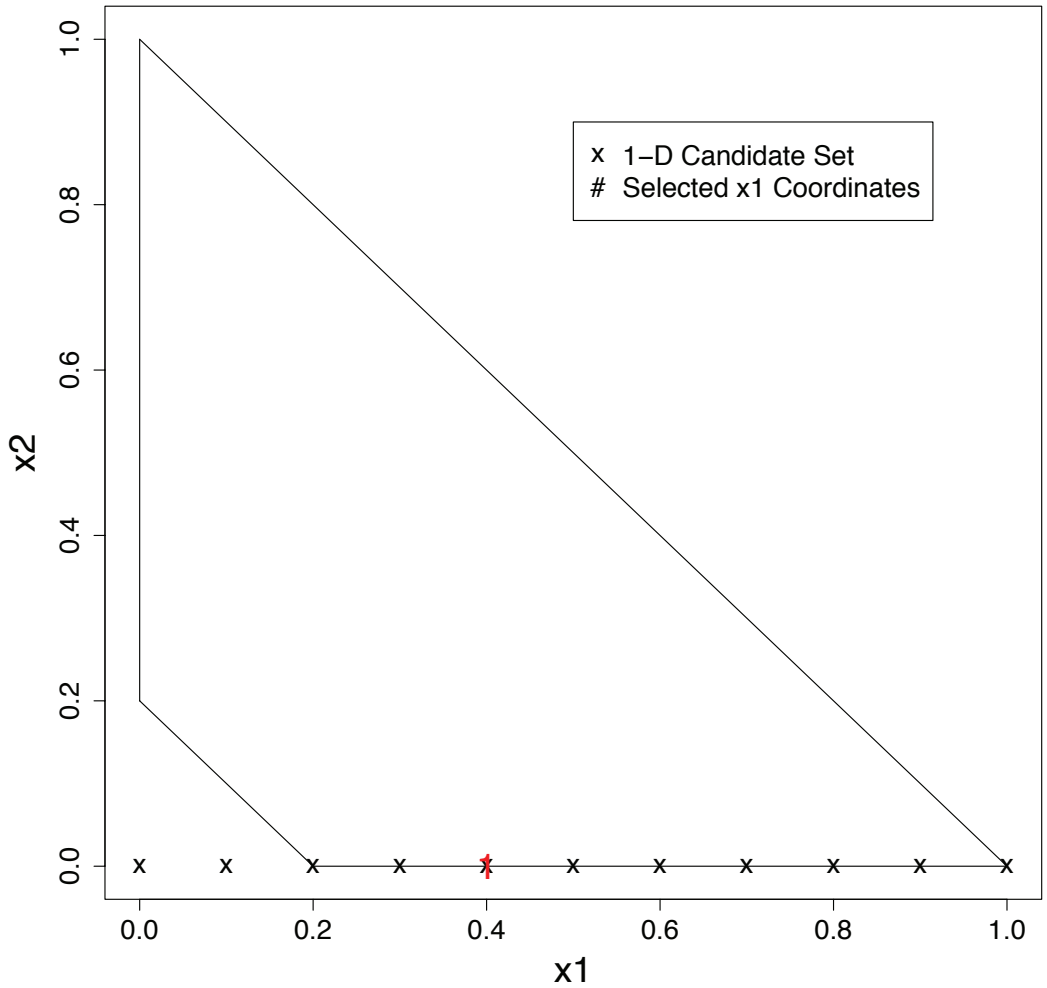
Multilayer Tool Coating:$n = 3$, $p = 2$ inputs

Step 1: build a 1-D candidate set for x_1



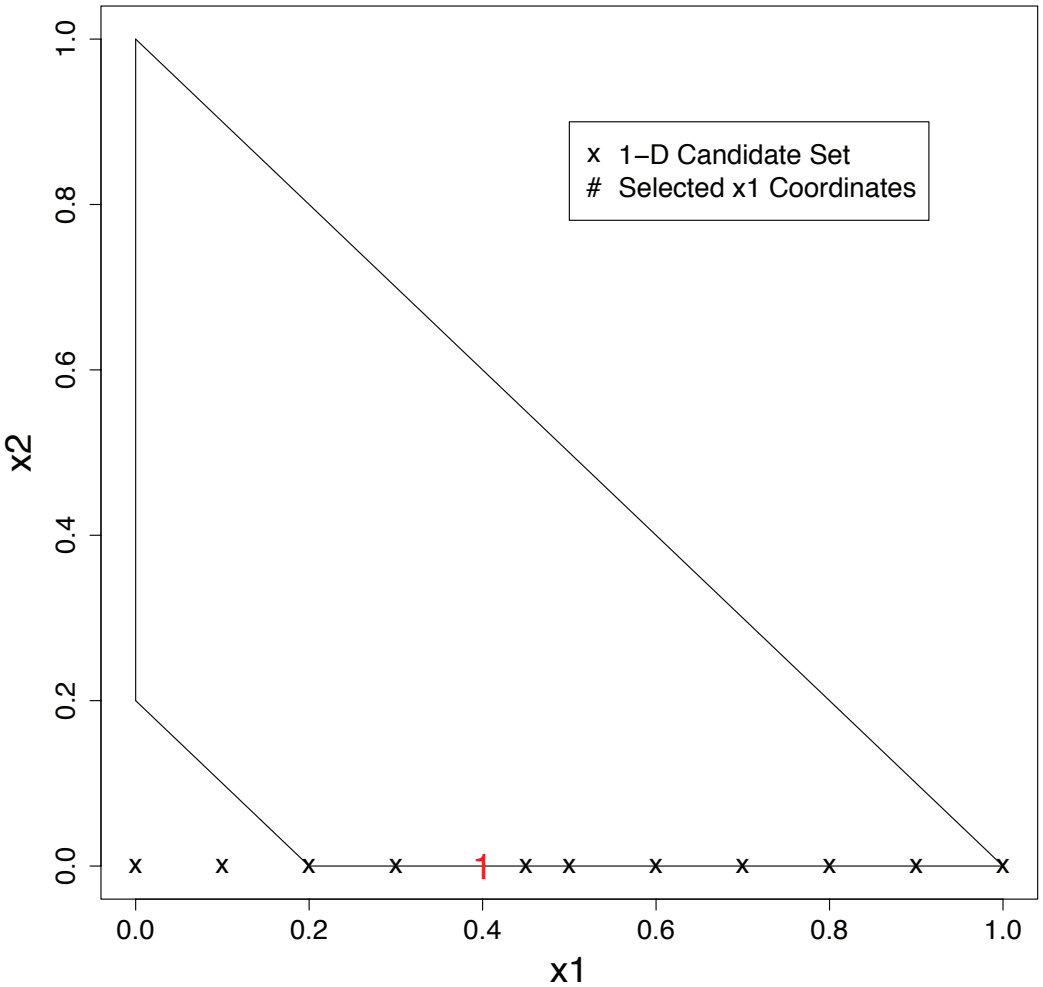
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Choose the first coordinate for x_1 randomly



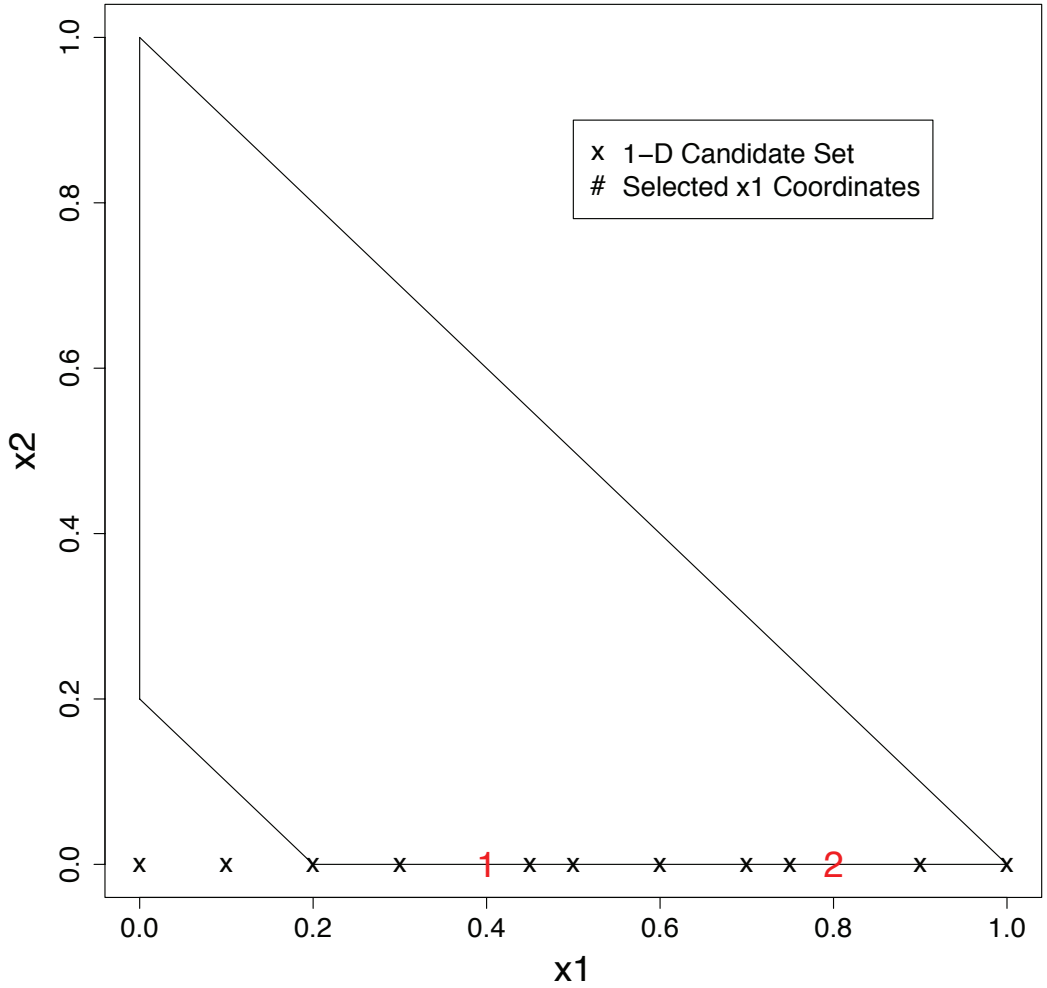
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Adapt the 1-D candidate set so that (i) previously selected $\mathbf{x_1}$ can not be chosen as a future x_1 choice and (ii) there are the same number of x_1 grid options as initially



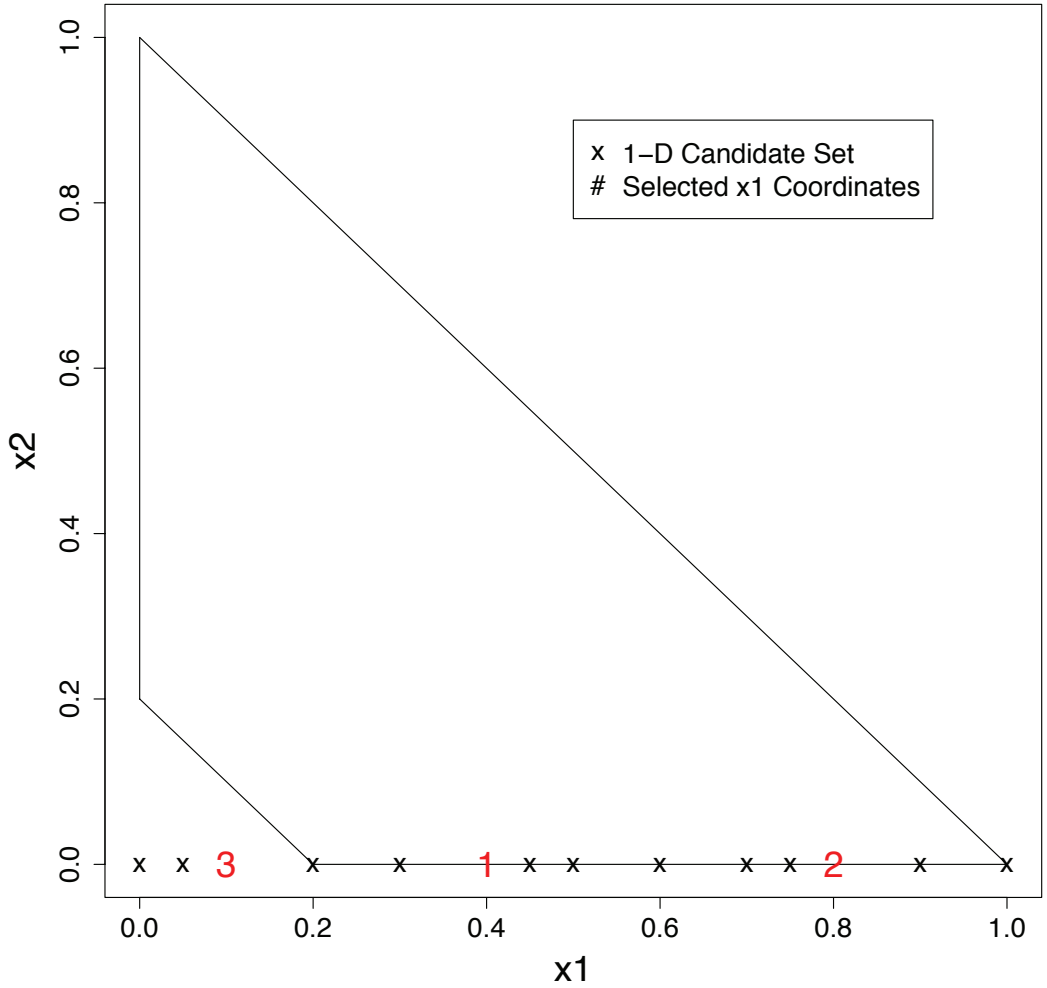
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Select 2^{nd} x_1 coordinate and adapt the candidate set



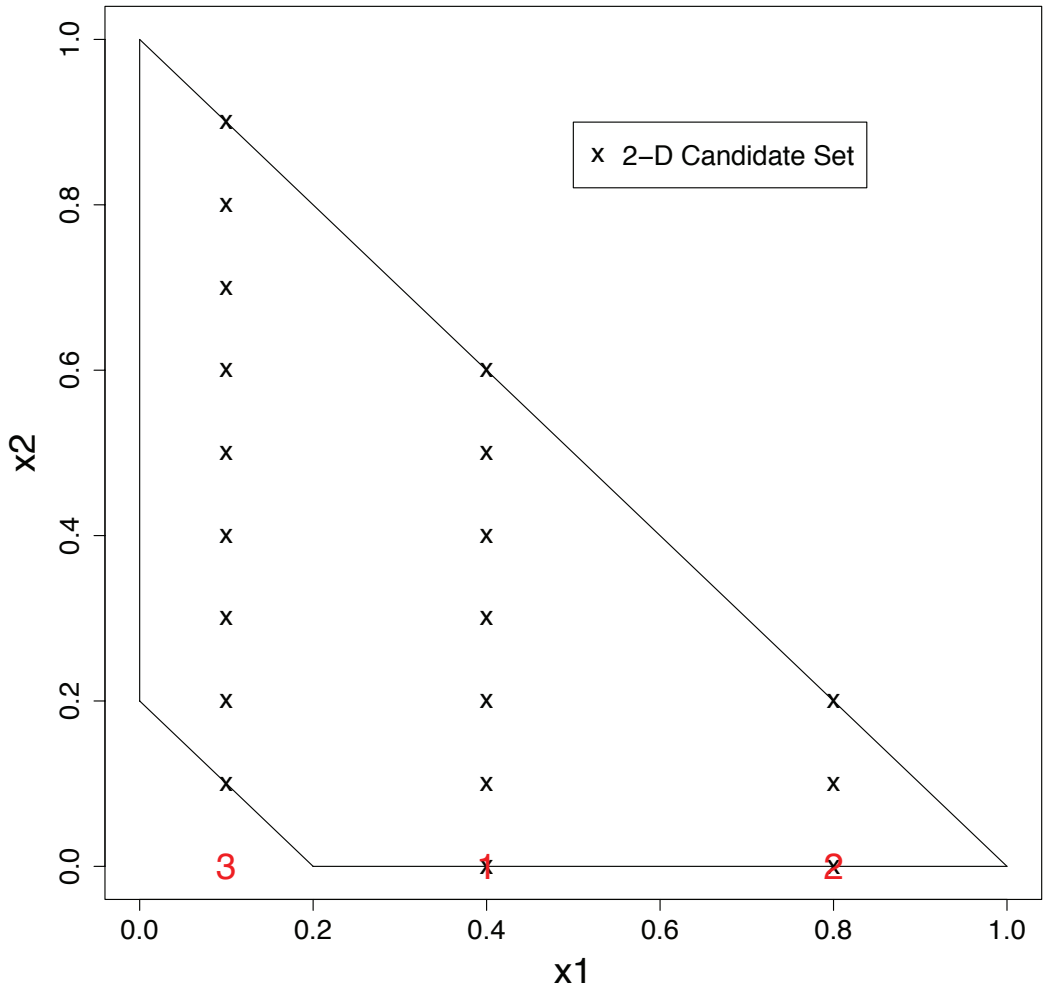
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Select 3rd x_1 coordinate and adapt the candidate set



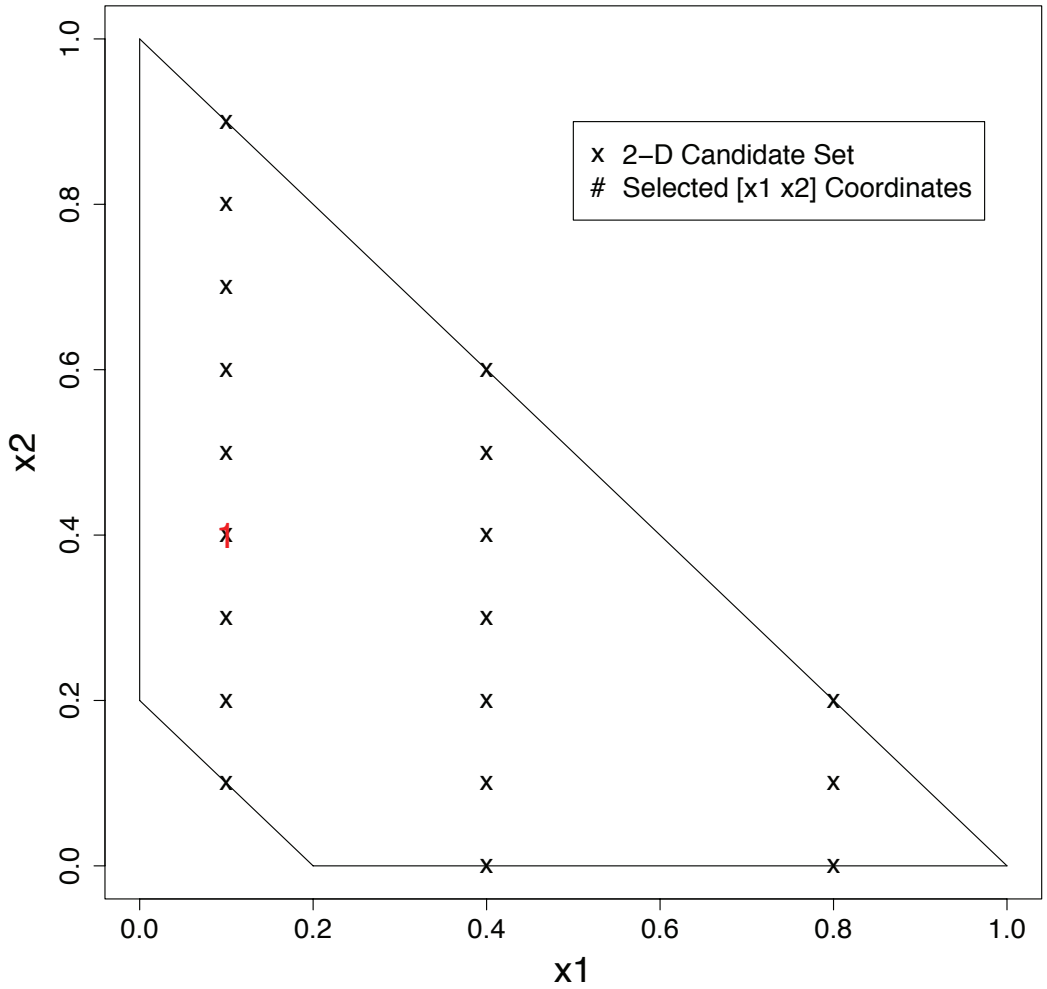
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Given the x_1 inputs, build the 2-D candidate set



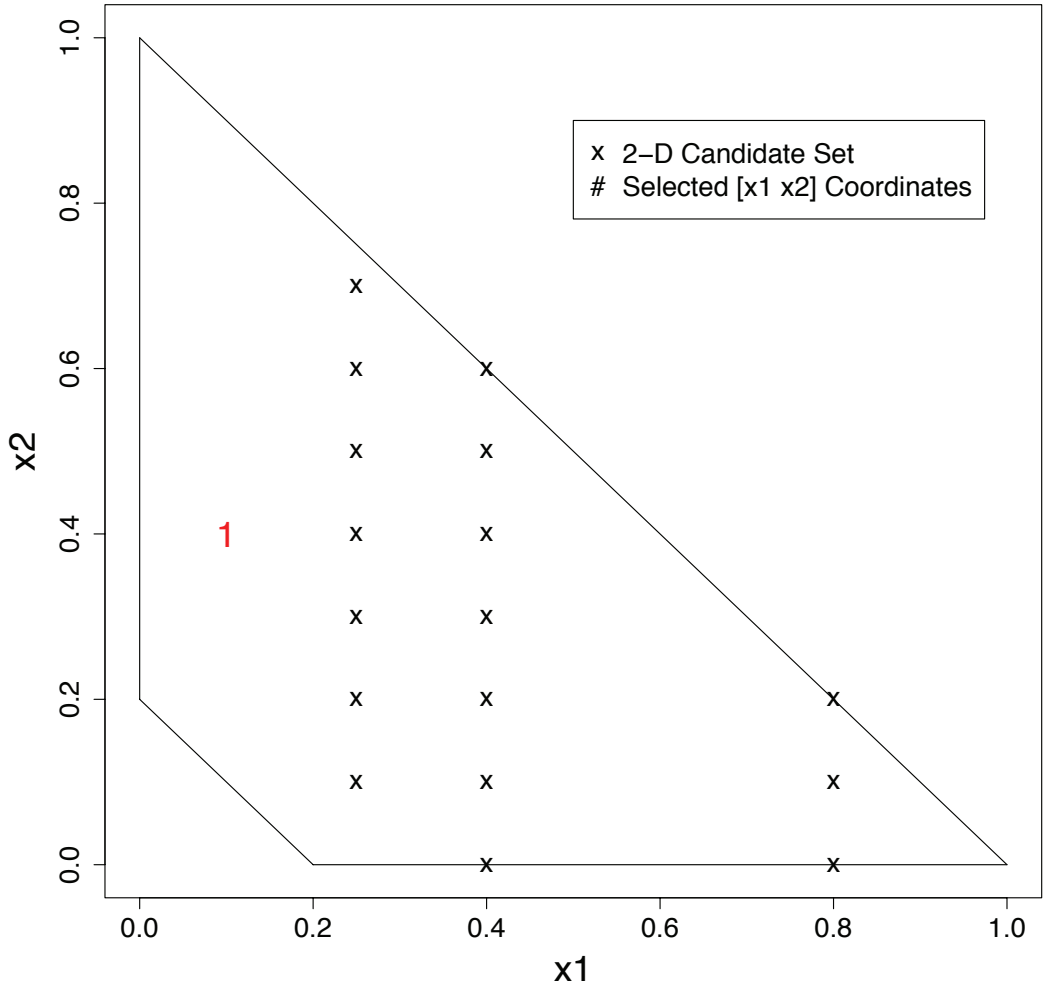
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Choose the first (x_1, x_2) input randomly



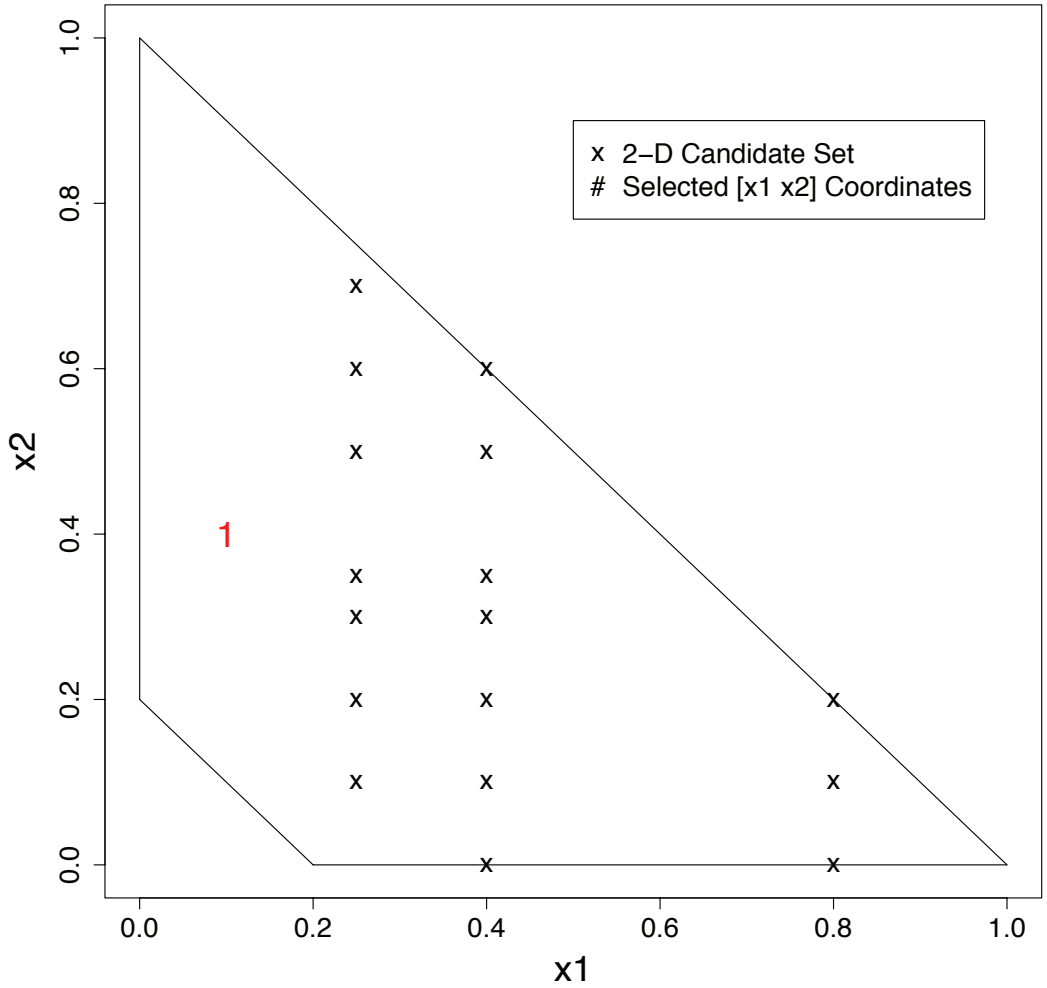
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Adapt the 2-D grid to prevent future design points from including selected x_1



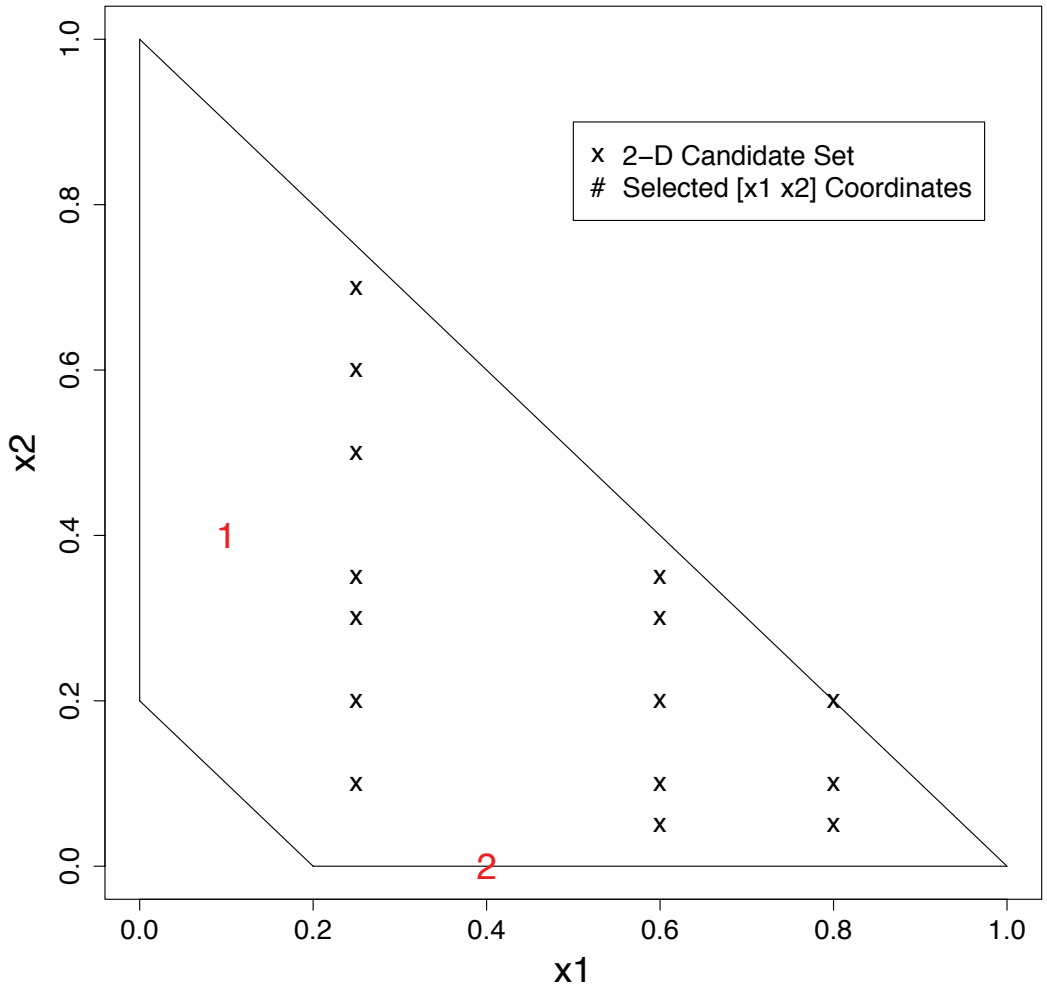
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Adapt the 2-D grid for x_2



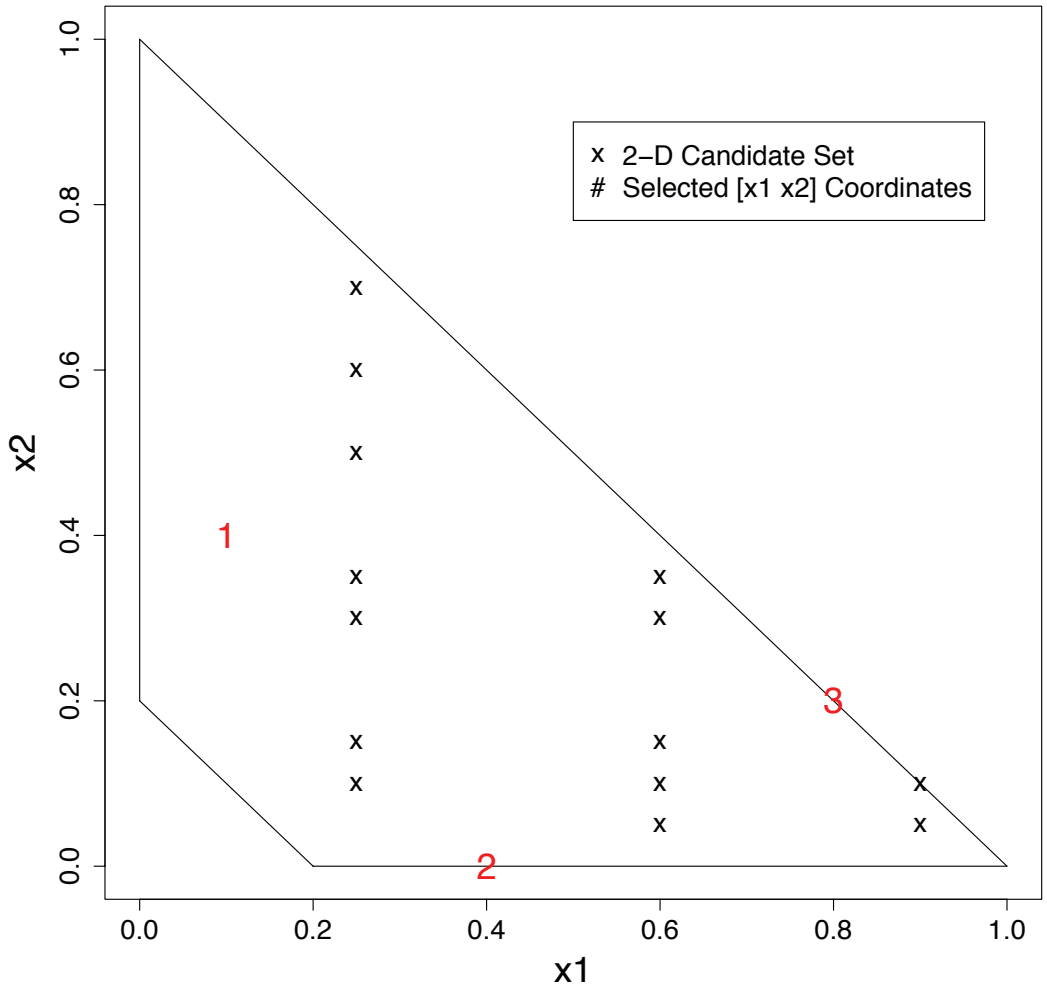
Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Select 2^{nd} point according to the α-space filling criterion; then adapt the candidate set



Algorithm Illustration: $n = 3$ with $p = 2$ inputs

Select 3rd point according to the α-space filling criterion; then adapt the candidate set



Tuning Parameters

CoNcaD algorithm uses two tuning parameters :

1. Grid density $u \equiv$ number of grid points over $[0, 1]$
   - Note: can scale the inputs to have marginal range [0, 1] by solving the linear programing (LP) problems

$$\min x_s \text{ subject to } \begin{cases} \boldsymbol{Ax} \leq \boldsymbol{b} \\ \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \end{cases}$$

Tuning Parameters

CoNcaD algorithm uses two tuning parameters :

1. Grid density $u \equiv$ number of grid points over $[0, 1]$
   - Note: can scale the inputs to have marginal range [0, 1] by solving the linear programing (LP) problems

$$\min x_s \text{ subject to } \begin{cases} \boldsymbol{Ax} \leq \boldsymbol{b} \\ \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \end{cases}$$

2. Excess Capacity $Q > n$ maximum number of points carried by the algorithm to counterbalance the myopic choice of additional inputs
   - recommend setting $u = 100$ and $Q = 3n$

Tuning Parameters

CoNcaD algorithm uses two tuning parameters :

1. Grid density $u \equiv$ number of grid points over $[0, 1]$
   - Note: can scale the inputs to have marginal range [0, 1] by solving the linear programing (LP) problems

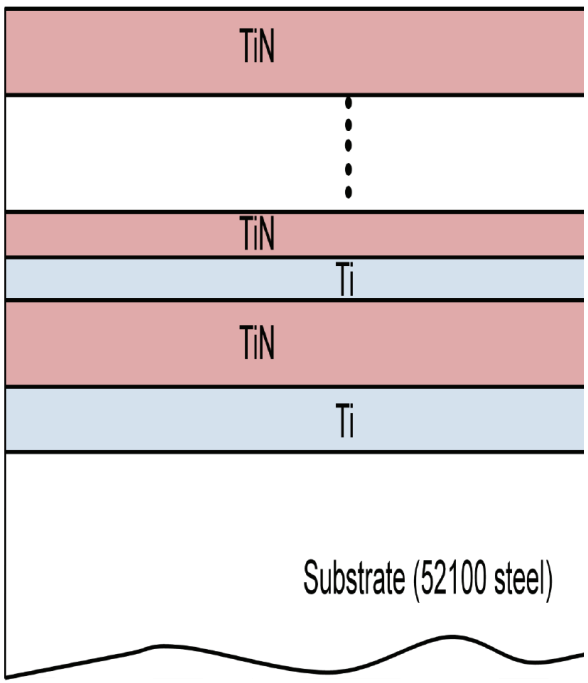
$$\min x_s \text{ subject to } \begin{cases} \boldsymbol{Ax} \leq \boldsymbol{b} \\ \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \end{cases}$$

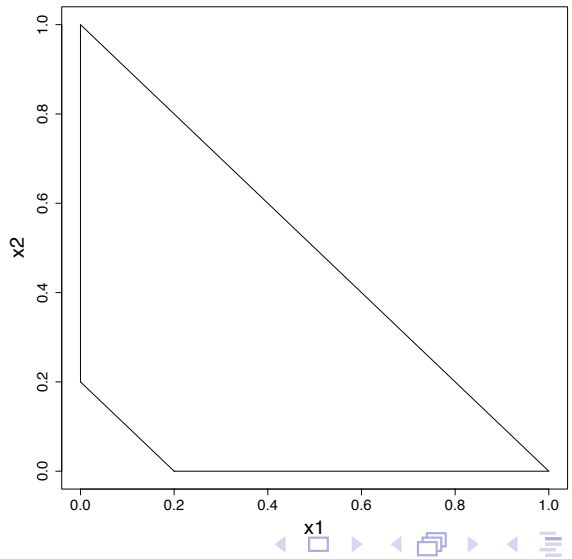
2. Excess Capacity $Q > n$ maximum number of points carried by the algorithm to counterbalance the myopic choice of additional inputs
   - recommend setting $u = 100$ and $Q = 3n$
3. Also recommend running the algorithm several times

Motivating Example 2: Multilayer Tool Coating

Nekkanty (2009) investigated the relationship between peak normal stress (cohesive cracks) and peak shear stress (delamination) as a function of number of coats and thickness. A coating design is defined by $(p, x_1, \ldots, x_p)$ where

$$p \in \{2, 4, 6, 8\} \text{ and } 2\ \mu m \leq \sum_{i=1}^{p} x_i \leq 6\ \mu m \text{ and } .25\ \mu m \leq x_i$$

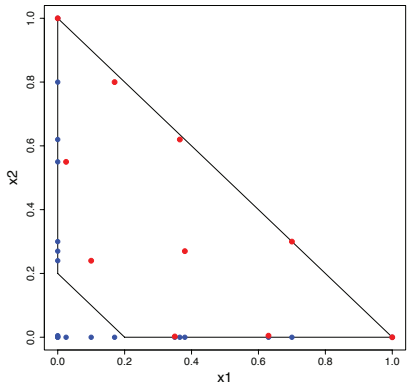




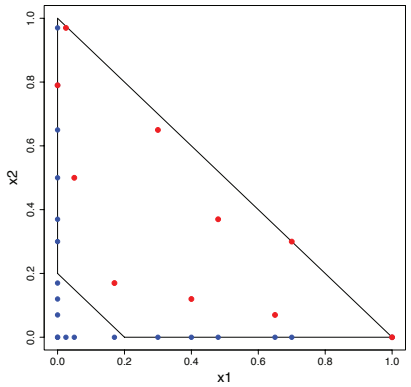
Design for Multilayer Tool Coating Experiment

$(n, p) = (10, 2)$ design using CoNcaD with $u = 100$, $Q = 3n$

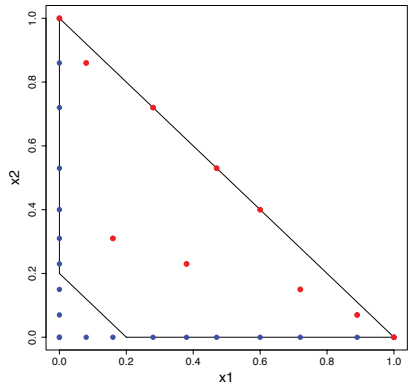
Criterion $\alpha R_{Mm} + (1 - \alpha) R_{ARD}$ with $\boldsymbol{J} = \{1, 2\}$



α = 1 (Mm)



α = 0.5 (Mm+ARD)

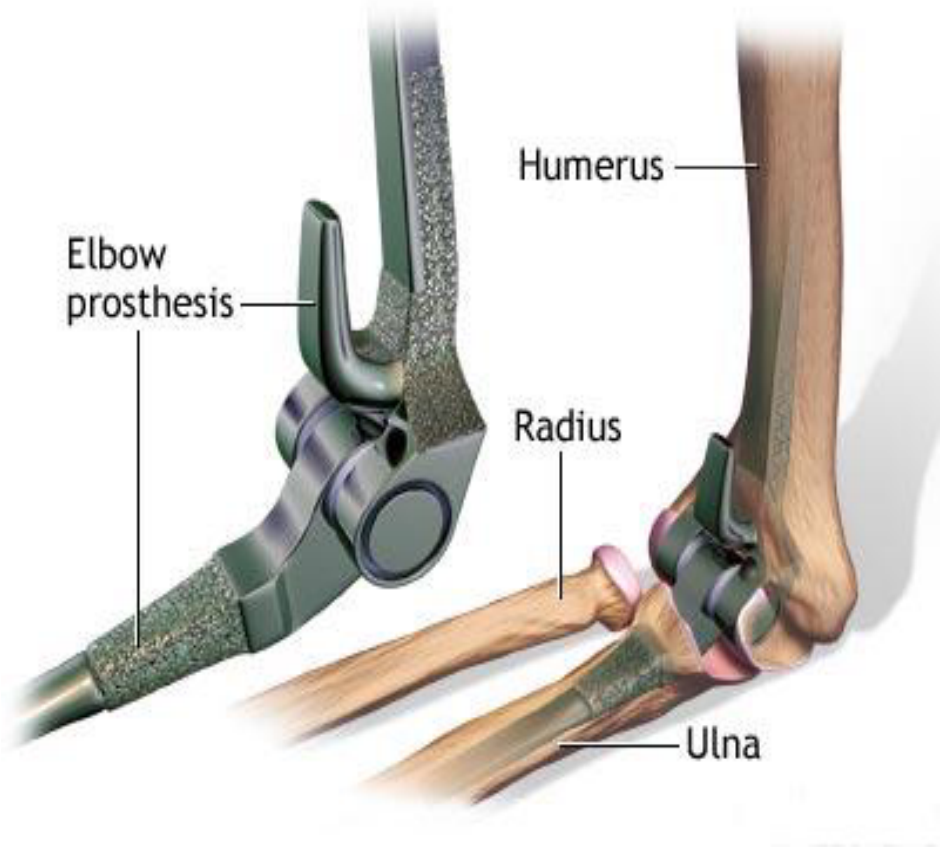


α = 0 (ARD)

α	1	0.5	0
mIPD	0.2625	0.1817	0.1304
ARD	12.5272	4.5303	3.8177

Motivating Example 1: Total elbow arthroplasty

Hayeck (2009) modeled compressive strains in the bone as a function of humeral implant position for a total elbow prosthesis



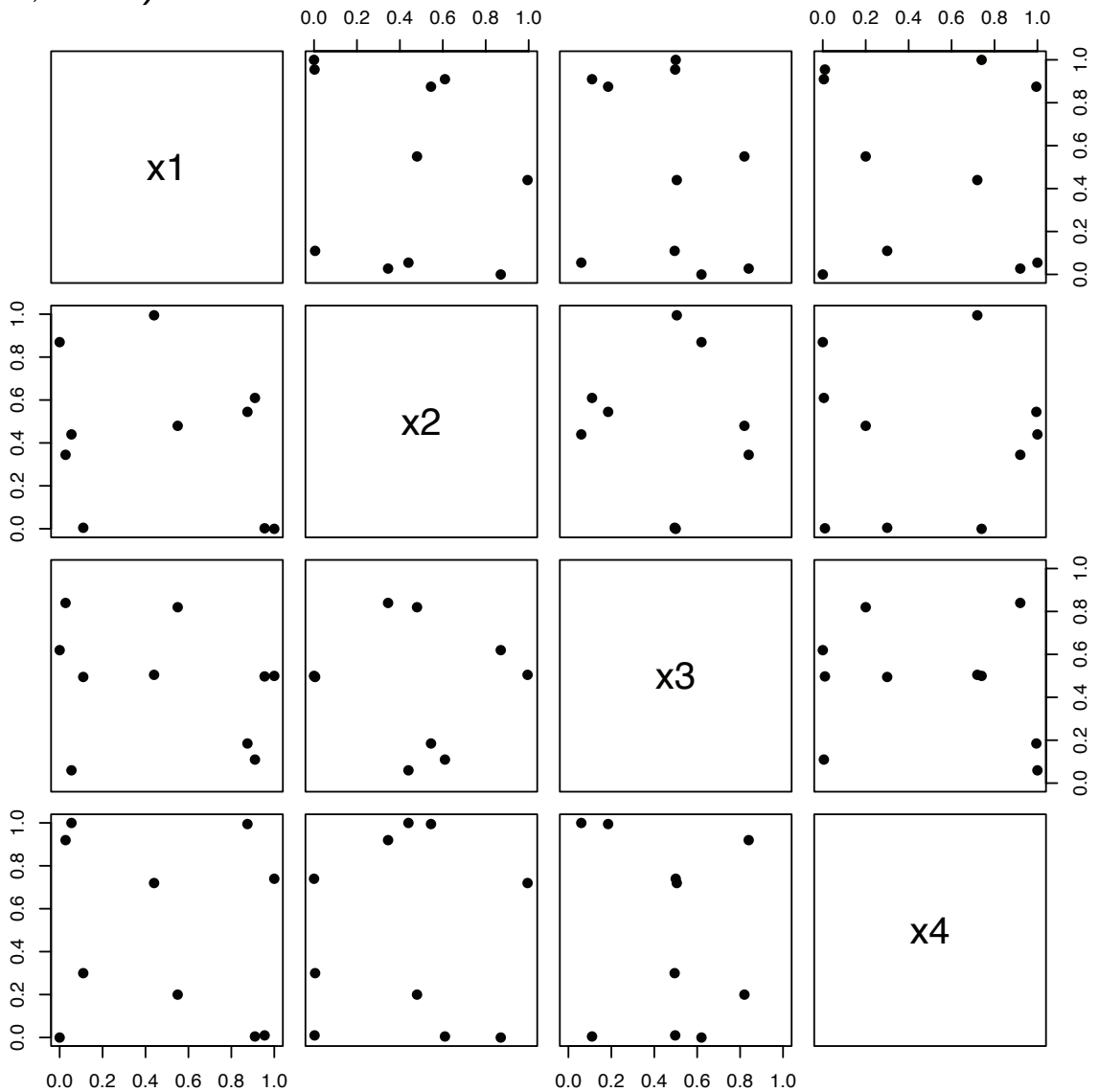
Motivating Example 1: Total elbow arthroplasty

- Used $p = 4$ variables to specify implant position
  - x_1 - tip displacement (mm).
  - x_2 - rotation of the implant axis about the lateral axis (°)
  - x_3 - rotation of the implant axis about the anterior axis (°) .
  - x_4 - rotation about the implant axis (°).
- Constraints: max tip displacement is 10 *mm*; rotation of implant axis is 10° about lateral axis at tip and 4° about anterior axis at tip; rotation about the implant axis is $\pm 15°$:

$$\begin{array}{rcl} 0 \leq & x_1 & \leq 10 \\ & 5x_2 + 2x_3 & \leq 10 \\ & -5x_2 + 2x_3 & \leq 10 \\ -10 \leq & 5x_2 + 2x_3 & \\ -10 \leq & -5x_2 + 2x_3 & \\ -15 \leq & x_4 & \leq 15 \end{array}$$

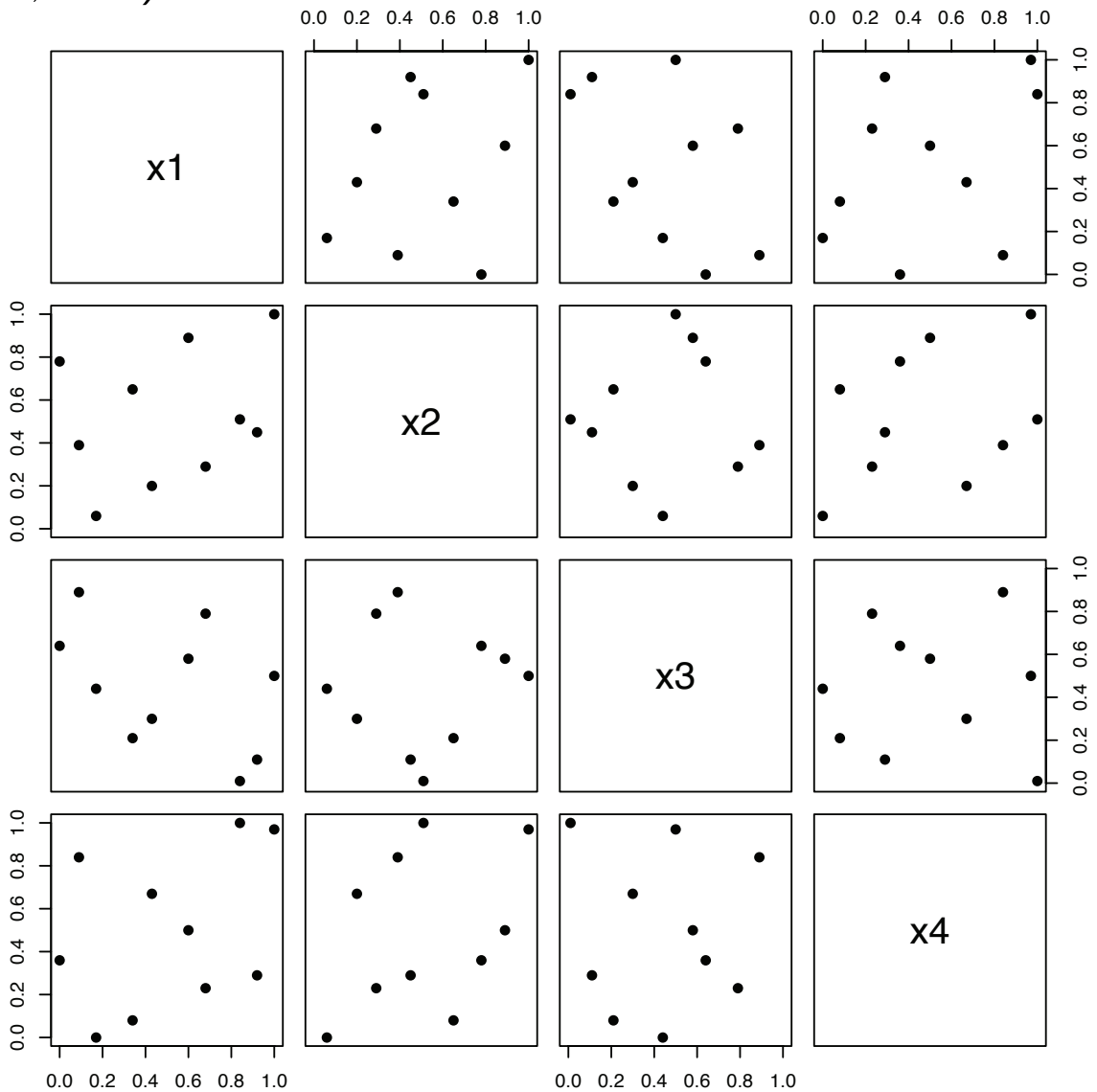
Elbow Implant Example: 2-D Projections

CoNcaD with $(n, p) = (10, 4)$, $\alpha = 1$ (Mm),
$(Q, u) = (30, 100)$



Elbow Implant Example: 2-D Projections

CoNcaD w/ $(n, p) = (10, 4)$, $\alpha = 0.5/\boldsymbol{J} = \{1, 2, 3, 4\}$, $(Q, u) = (30, 100)$



Algorithm Highlights

1. The CoNcaD algorithm produces space-filling (in the sense of Mm-ARD-Mm/ARD), non-collapsing designs

Algorithm Highlights

1. The CoNcaD algorithm produces space-filling (in the sense of Mm-ARD-Mm/ARD), non-collapsing designs
2. Scales up well in p

Algorithm Highlights

1. The CoNcaD algorithm produces space-filling (in the sense of Mm-ARD-Mm/ARD), non-collapsing designs
2. Scales up well in p
3. Because of the random selection to start the construction of each $\boldsymbol{X}^{(s)}$, multiple runs of the algorithm can give different designs
   $\rightarrow$ based on our studies of the algorithms performance we recommend setting $(u, Q) = (100, 3n)$ to minimize the impact of random selection; also run the algorithm multiple times (at least 10)

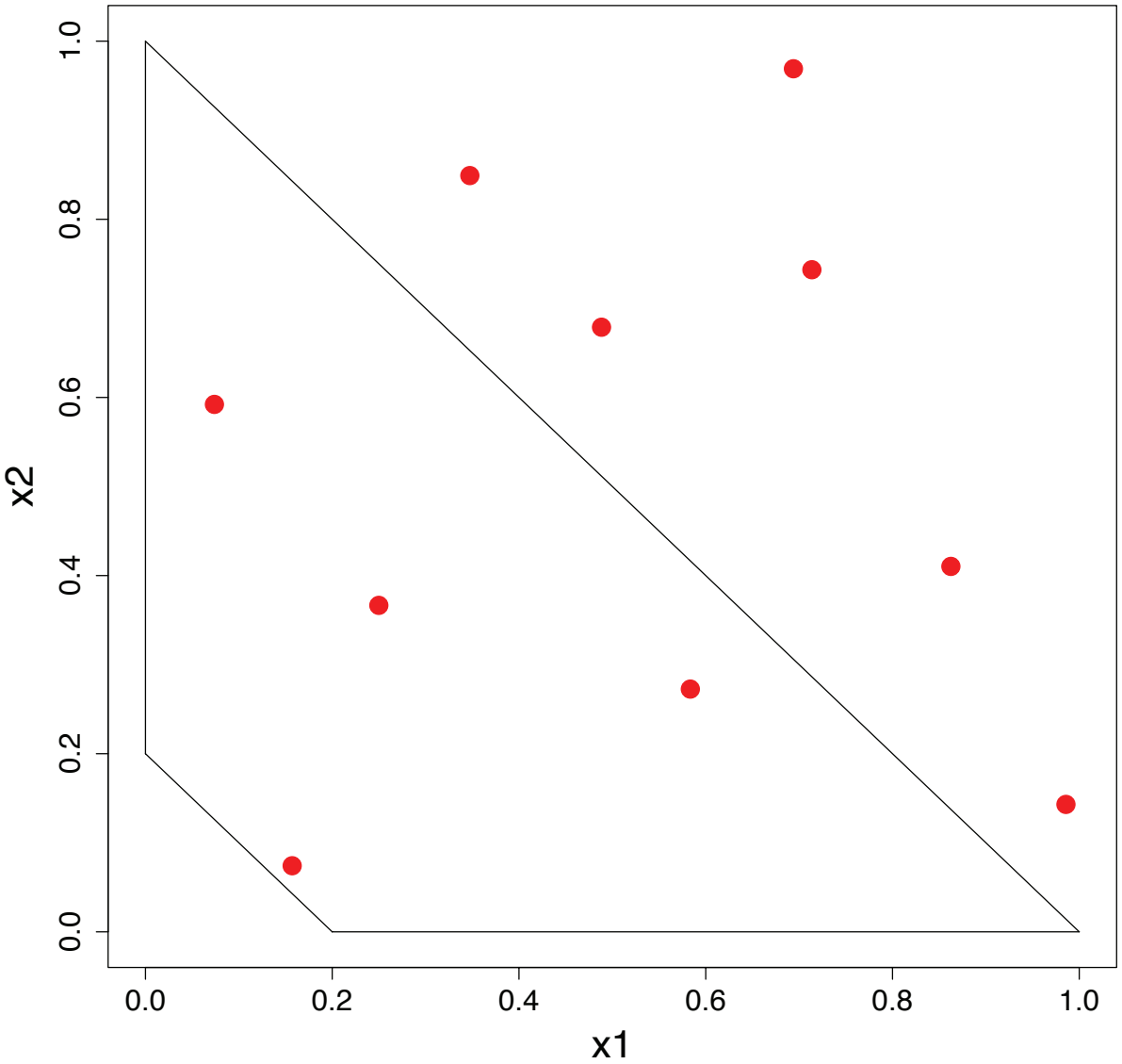
Algorithm Highlights

1. The CoNcaD algorithm produces space-filling (in the sense of Mm-ARD-Mm/ARD), non-collapsing designs
2. Scales up well in p
3. Because of the random selection to start the construction of each $\boldsymbol{X}^{(s)}$, multiple runs of the algorithm can give different designs
   $\rightarrow$ based on our studies of the algorithms performance we recommend setting $(u, Q) = (100, 3n)$ to minimize the impact of random selection; also run the algorithm multiple times (at least 10)
4. Algorithm can be extended to augment given designs.

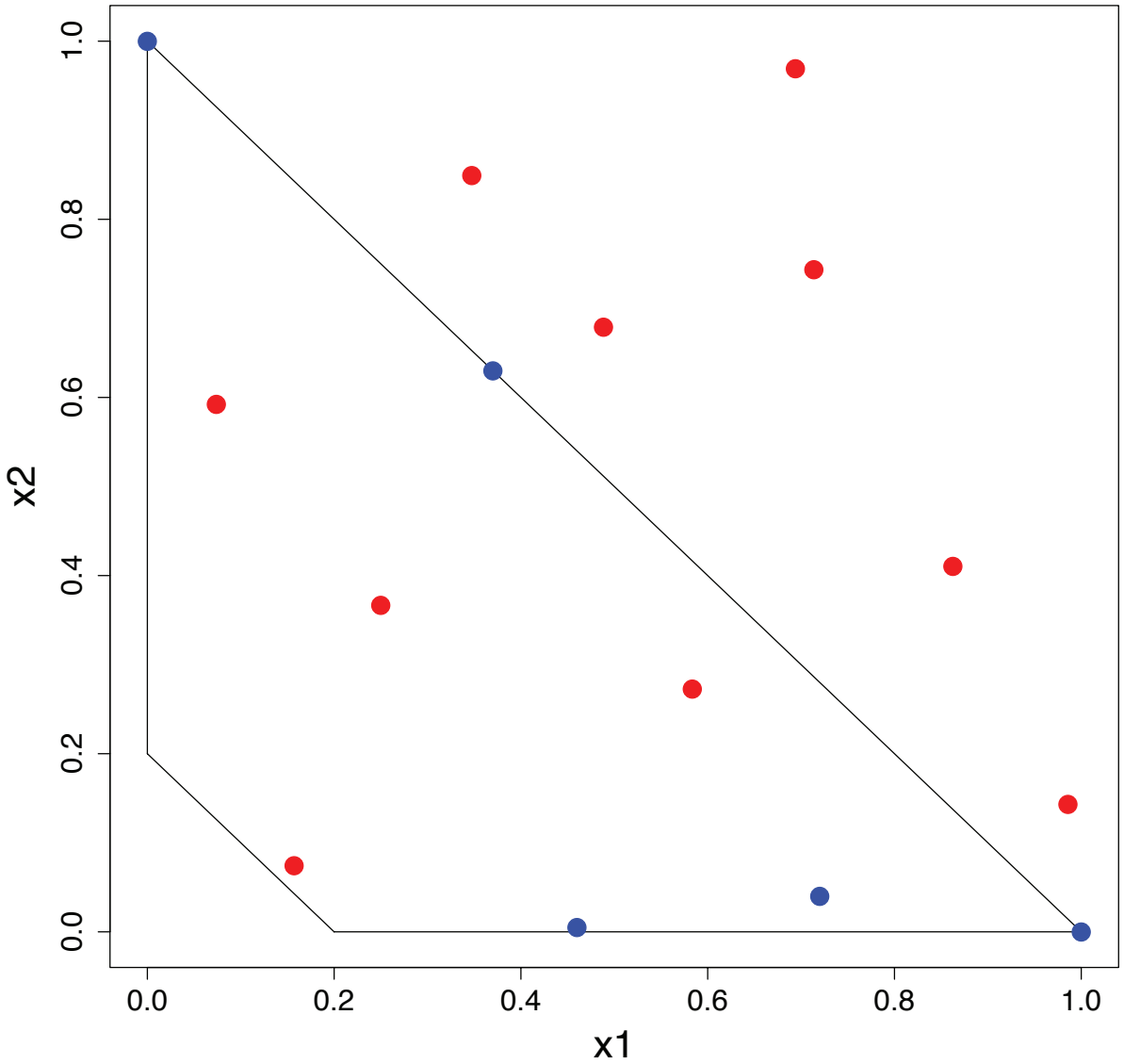
Algorithm Highlights

1. The CoNcaD algorithm produces space-filling (in the sense of Mm-ARD-Mm/ARD), non-collapsing designs
2. Scales up well in p
3. Because of the random selection to start the construction of each $\boldsymbol{X}^{(s)}$, multiple runs of the algorithm can give different designs
   $\rightarrow$ based on our studies of the algorithms performance we recommend setting $(u, Q) = (100, 3n)$ to minimize the impact of random selection; also run the algorithm multiple times (at least 10)
4. Algorithm can be extended to augment given designs.
5. Algorithm can be extended to non-polygonal regions

Extension 1: Nested Design



Extension 1: Nested Design



Extension Two- and Ten-Dimensional Ball Examples

Constraint: $0 \leq \sum_{i=1}^{p} x_i^2 \leq 1$ for $p = 2, 10$; Mm design:

p	n	SFDP*[a]	SFDP**[a]	Trosset[a]	CoNcaD
2	10	0.3587	0.3522	0.2554	0.3313
	100	NA[b]	0.0486	0.0660	0.0811
	200	NA[b]	0.0259	0.0070	0.0546
10	10	1.4142	1.2167	0.5017	1.1160
	100	NA[b]	0.6248	0.2753	0.5712
	200	NA[b]	0.5235	0.2559	0.4977

[a]The designs were obtained from Stinstra et. al (2003) and need not be non-collapsing

[b]The number of constraints was too large

Thank you!

Algorithm Statement

1. Given the desired design (n, p) and criterion $(\alpha, \boldsymbol{J} \subseteq \{1, 2, \ldots, p\})$, select Q, u
   Construct the grid $\boldsymbol{m} = (0/u, 1/u, \ldots, 1)$ and define a candidate set $\boldsymbol{C}^{(1)} = \boldsymbol{m}$.
2. At step s $(s \geq 2)$, construct the tentative $Qu \times s$ candidate set $\boldsymbol{C}^{(s)}$ used to construct the $Q \times s$ design $\boldsymbol{X}^{(s)}$ by

$$\boldsymbol{C}^{(s)} = \begin{pmatrix} \boldsymbol{X}^{(s-1)} & m_1 \mathbf{1}_Q \\ \boldsymbol{X}^{(s-1)} & m_2 \mathbf{1}_Q \\ \vdots & \vdots \\ \boldsymbol{X}^{(s-1)} & m_u \mathbf{1}_Q \end{pmatrix}$$

where $\mathbf{1}_Q$ is a $Q \times 1$ vector of ones so that each m_i is appended to a copy of each row of $\boldsymbol{X}^{(s-1)}$.

Algorithm Statement

3. Eliminate the rows of $\boldsymbol{C}^{(s)}$ if they do not satisfy the constraint $\boldsymbol{Ax} \leq \boldsymbol{b}$.
4. Choose the first row for the $Q \times s$ design matrix $\boldsymbol{X}^{(s)}$ from $\boldsymbol{C}^{(s)}$ randomly.
5. (a) Adapt $\boldsymbol{C}^{(s)}$ to implement non-collapsingness by modifying the coordinates of $\boldsymbol{C}^{(s)}$ which coincide with coordinates of all previously selected points.

   (b) Select rows 2 to Q of $\boldsymbol{X}^{(s)}$ from $\boldsymbol{C}^{(s)}$ which when, appended to $\boldsymbol{X}^{(s)}$, to minimize

$$\alpha \text{Rank}_{mIPD} + (1 - \alpha)\text{Rank}_{ARD}.$$

   readapting $\boldsymbol{C}^{(s)}$ after each row is added (update formulas)

Algorithm Statement

6. Continue until there are Q rows in $\boldsymbol{X}^{(s)}$.

7. At step $s = p$, select only n rows using (5) so that the final design is $n \times p$ points.