Two-level orthogonal screening designs with 80, 96 and 112 runs: Construction and evaluation

Alan R. Vazquez

University of California, Los Angeles

alanrvazquez@stat.ucla.edu

Quality and Productivity Research Conference

July 26-30, 2021

Funded by the Research Foundation – Flanders, Belgium

Outline

1. Tuberculosis inhibition experiment

2. Criteria to evaluate designs

3. Construction method

4. Results and discussion

Tuberculosis inhibition experiment

- Silva et al. (2016) conducted a study to develop a treatment that maximizes the percentage of inhibition of tuberculosis in infected human cells.
- The first stage of the study involved a screening experiment.
- 14 factors (drugs) at two levels (presence or absence).

Goal:

Identify the influential main effects and two-factor interactions of the factors.

For the experiment, the researchers chose an attractive two-level design with 14 factors and 128 runs.

It is a **strength-4** orthogonal design (Hedayat et al. 1999) which allows to estimate all main effects and all two-factor interactions with full precision.

In the end, however, only 8 main effects and 6 two-factor interactions were active.

Research question:

Could we have identified the active effects with a smaller design?

Literature review

Strength-3 orthogonal designs. Properties:

1. Main effects are **not correlated** with each other **nor** with two-factor interactions.
2. Pairs of two-factor interactions can be correlated.
3. Exist for run sizes which are multiples of 8.

Example:

- Regular fractional factorial designs of resolution IV.
- Folded-over Plackett-Burman designs (nonregular).

Catalogs of strength-3 designs with 16, 24, 32, 40 and 48 runs and up to 24 factors are available (Schoen et al., 2010).

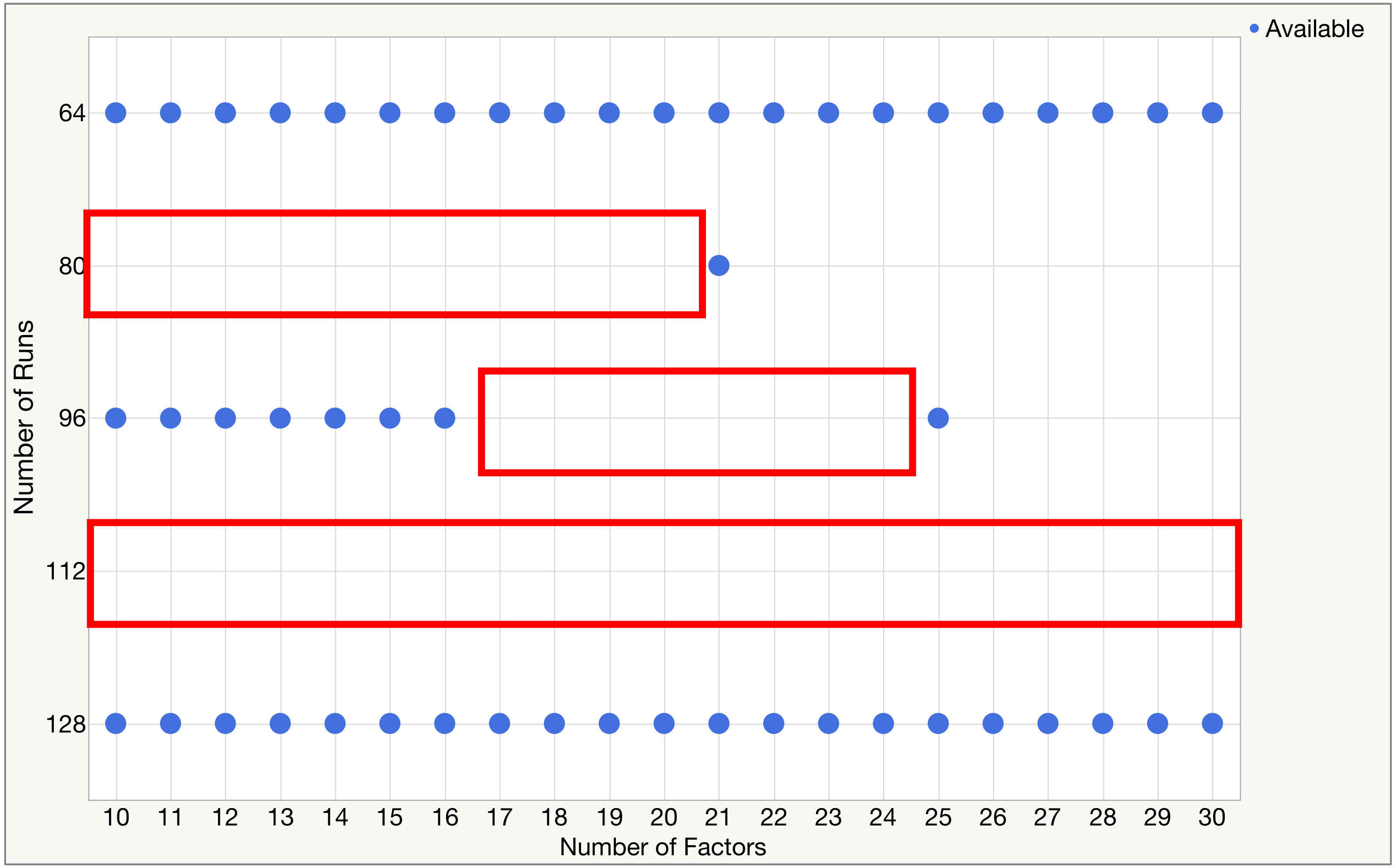
Available strength-3 designs with run sizes from 64 to 128

Chen et al. (1993); Xu & Wong (2007); Cheng et al. (2008); Vazquez & Xu (2019); Vazquez et al. (2019).

Cheng et al. (2008).

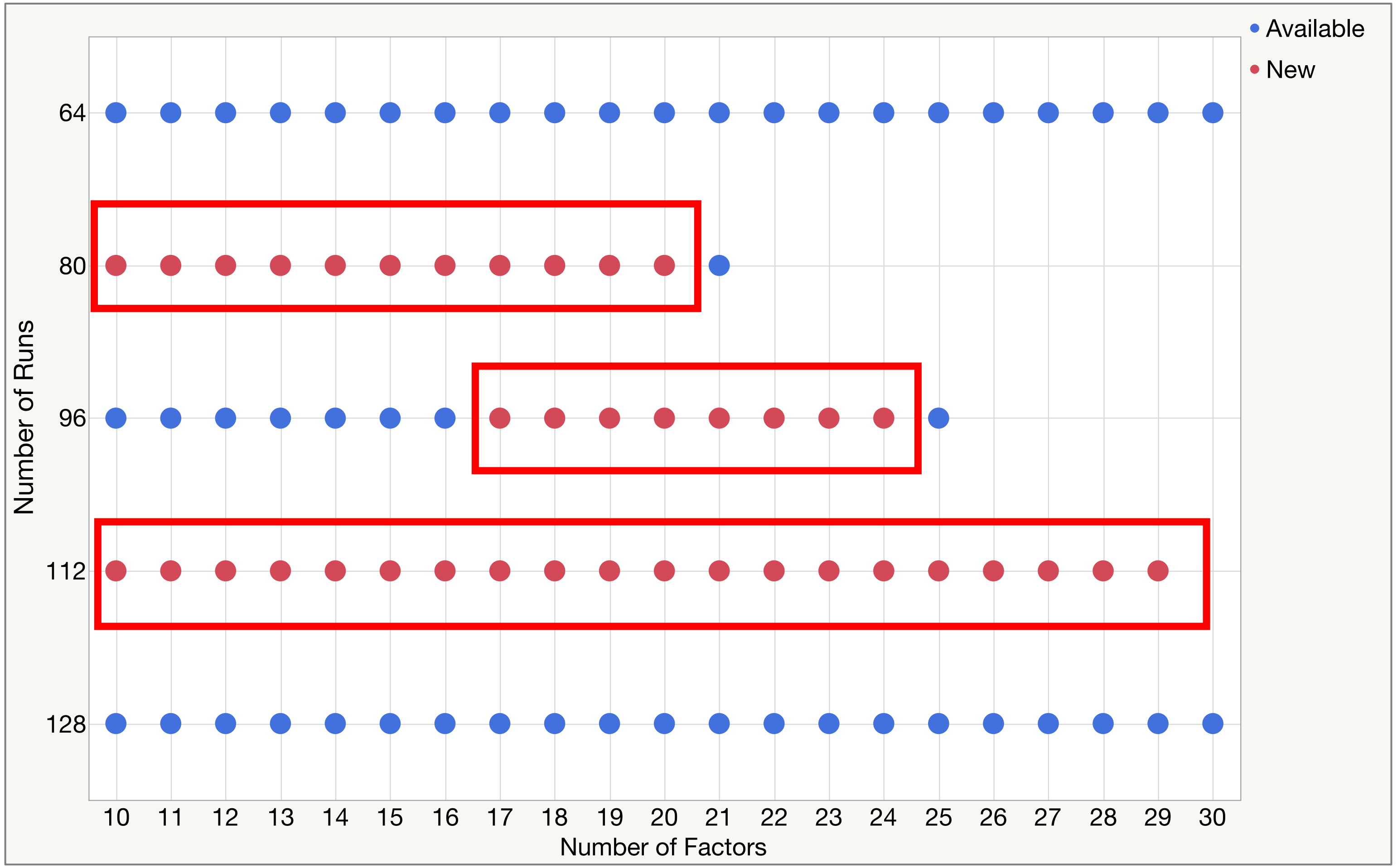
Cheng et al. (2008); Vazquez & Xu (2019).

Ryan & Bulutoglu (2010); Xu & Wong (2007); Vazquez & Xu (2019); Vazquez et al. (2019)



This talk

New strength-3 orthogonal designs which fill the gaps between the available designs.



Outline

1. Tuberculosis inhibition experiment
2. Criteria to evaluate designs
3. Construction method
4. Results and discussion

Evaluating two-level strength-3 designs

Example: Compare designs with 32 runs and 10 factors.

Regular resolution-IV design

1	2	3	4	5	6	7	8	9	10
−1	−1	−1	−1	−1	−1	−1	−1	1	1
−1	−1	−1	−1	1	−1	−1	1	−1	−1
−1	−1	−1	1	−1	−1	1	−1	−1	−1
−1	−1	−1	1	1	−1	1	1	1	1
−1	−1	1	−1	−1	1	−1	−1	−1	−1
−1	−1	1	−1	1	1	−1	1	1	1
−1	−1	1	1	−1	1	1	−1	1	1
−1	−1	1	1	1	1	1	1	−1	−1
−1	1	−1	−1	−1	1	1	1	1	−1
−1	1	−1	−1	1	1	1	−1	−1	1
−1	1	−1	1	−1	1	−1	1	−1	1
−1	1	−1	1	1	1	−1	−1	1	−1
−1	1	1	−1	−1	−1	1	1	−1	1
−1	1	1	−1	1	−1	1	−1	1	−1
−1	1	1	1	−1	−1	−1	1	1	−1
−1	1	1	1	1	−1	−1	−1	−1	1
1	−1	−1	−1	−1	1	1	1	−1	1
1	−1	−1	−1	1	1	1	−1	1	−1
1	−1	−1	1	−1	1	−1	1	1	−1
1	−1	−1	1	1	1	−1	−1	−1	1
1	−1	1	−1	−1	−1	1	1	1	−1
1	−1	1	−1	1	−1	1	−1	−1	1
1	−1	1	1	−1	−1	−1	1	−1	1
1	−1	1	1	1	−1	−1	−1	1	−1
1	1	−1	−1	−1	−1	−1	−1	−1	−1
1	1	−1	−1	1	−1	−1	1	1	1
1	1	−1	1	−1	−1	1	−1	1	1
1	1	−1	1	1	−1	1	1	−1	−1
1	1	1	−1	−1	1	−1	−1	1	1
1	1	1	−1	1	1	−1	1	−1	−1
1	1	1	1	−1	1	1	−1	−1	−1
1	1	1	1	1	1	1	1	1	1

Wu & Hamada (2009)

Nonregular strength-3 design

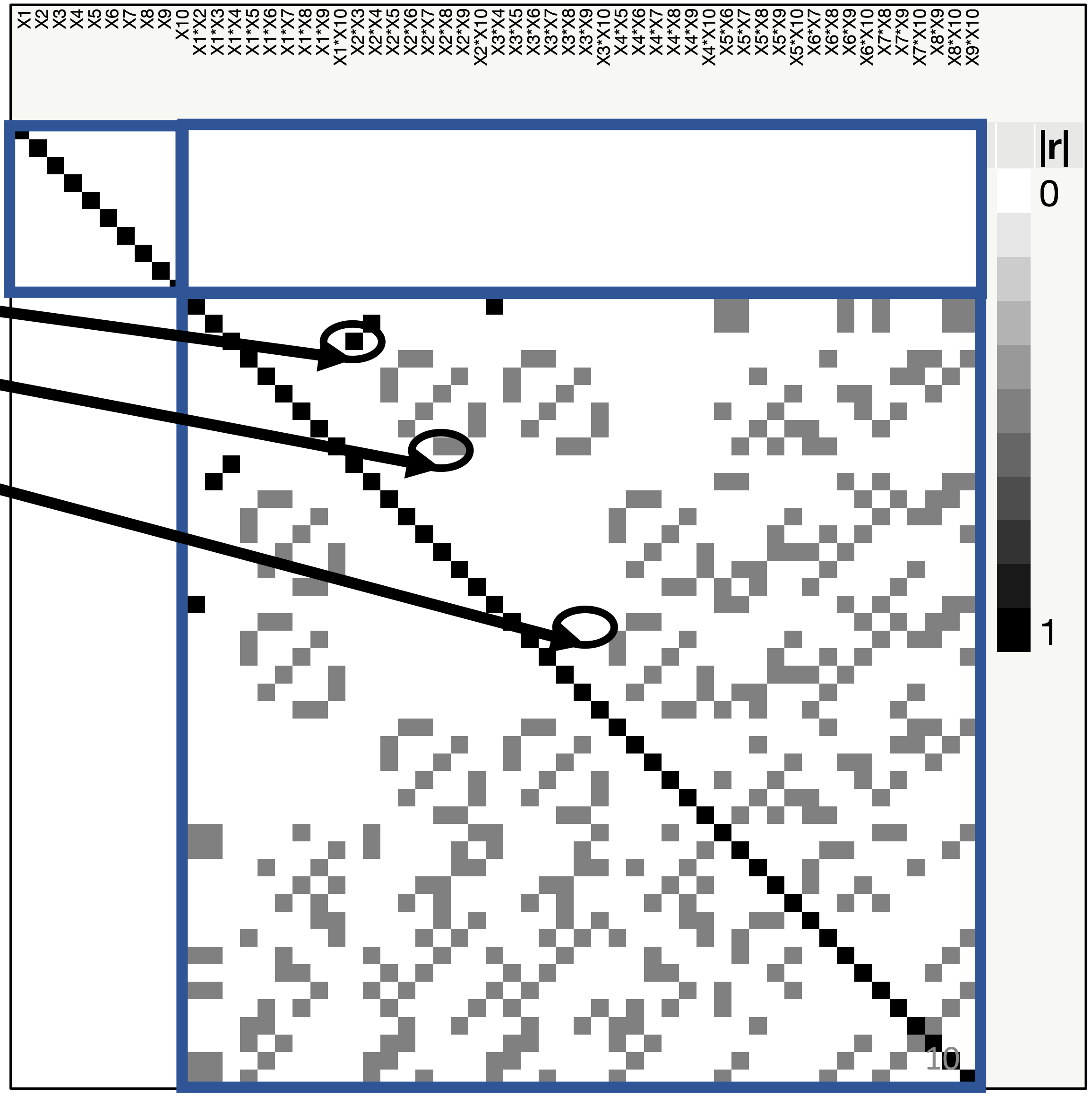
1	2	3	4	5	6	7	8	9	10
−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	−1	1	1	1	1
−1	−1	−1	−1	1	1	−1	−1	1	1
−1	−1	−1	−1	1	1	1	1	−1	−1
−1	−1	1	1	−1	−1	−1	−1	−1	−1
−1	−1	1	1	−1	1	−1	1	1	1
−1	−1	1	1	1	−1	1	1	−1	1
−1	−1	1	1	1	1	1	−1	1	−1
−1	1	−1	1	−1	−1	1	−1	1	1
−1	1	−1	1	−1	1	1	1	−1	−1
−1	1	−1	1	1	−1	−1	1	1	−1
−1	1	−1	1	1	1	−1	−1	−1	1
−1	1	1	−1	−1	1	−1	1	1	−1
−1	1	1	−1	−1	1	1	−1	−1	1
−1	1	1	−1	1	−1	−1	1	−1	1
−1	1	1	−1	1	−1	1	−1	1	−1
1	−1	−1	1	−1	1	−1	1	−1	1
1	−1	−1	1	−1	1	1	−1	1	−1
1	−1	−1	1	1	−1	−1	1	1	−1
1	−1	−1	1	1	−1	1	−1	−1	1
1	−1	1	−1	−1	−1	1	1	1	−1
1	−1	1	−1	−1	1	1	−1	−1	1
1	−1	1	−1	1	−1	−1	−1	1	1
1	−1	1	−1	1	1	−1	1	−1	−1
1	1	−1	−1	−1	−1	−1	1	−1	1
1	1	−1	−1	−1	1	−1	−1	1	−1
1	1	−1	−1	1	−1	1	−1	−1	−1
1	1	−1	−1	1	1	1	1	1	1
1	1	1	1	−1	−1	−1	−1	1	1
1	1	1	1	−1	−1	1	1	−1	−1
1	1	1	1	1	1	−1	−1	−1	−1
1	1	1	1	1	1	1	1	1	1

Schoen & Mee (2010)

Color map on correlations

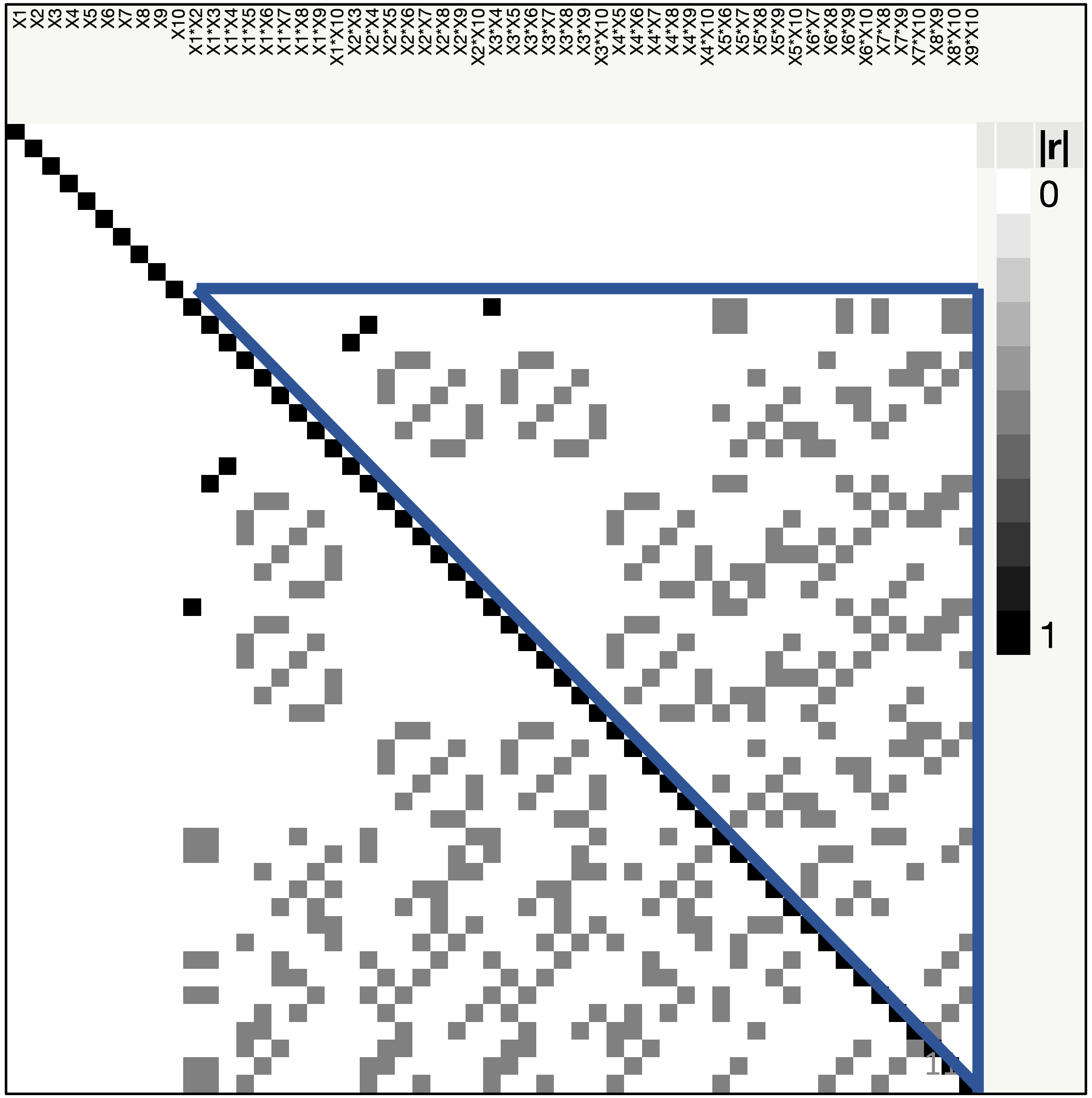
For 32-run designs, the absolute correlations can have values of 0, 0.5 or 1 (Deng & Tang, 1999).

Nonregular strength-3 design



How can we summarize the correlation between pairs of two-factor interactions in strength-3 designs?

Nonregular strength-3 design



Two statistical criteria:

- **The F_4 vector**
- **The B_4 value**

(Deng & Tang, 1999; Tang & Deng 1999)

How can we summarize the correlation between pairs of two-factor interactions in strength-3 designs?

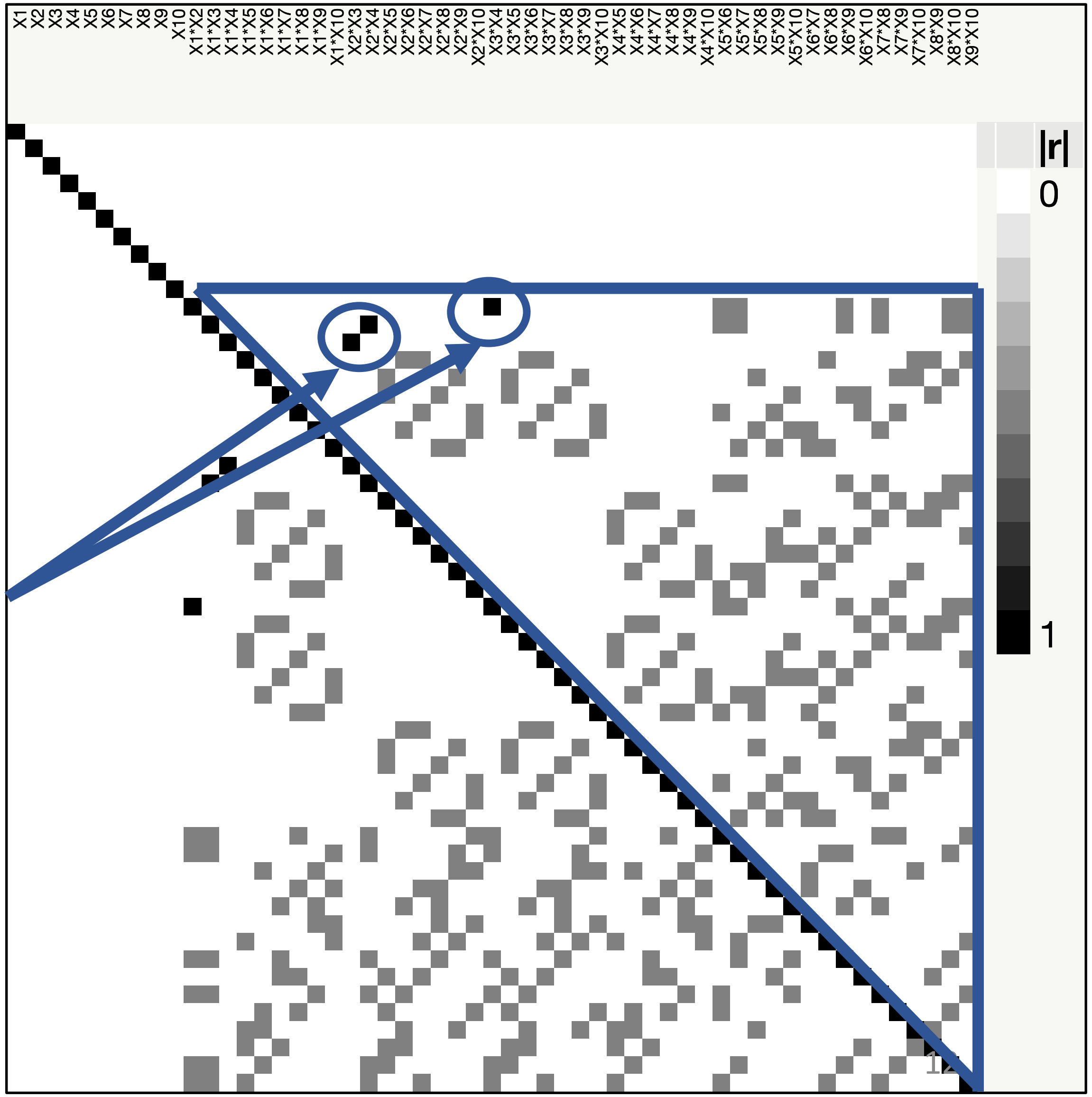
The F_4 vector has entries equal to the frequencies for the possible absolute correlation values between pairs of interactions, divided by three (Deng & Tang, 1999).

- $F_4(1, 0.5) = (1, 62)$
- 3 pairs with an abs. correlation of 1.
- 186 pairs with an abs. correlation of 0.5.

A good strength-3 design sequentially minimizes the F_4 vector.

Nonregular strength-3 design

(X_1X_2, X_3X_4)
(X_1X_3, X_2X_4)
(X_1X_4, X_2X_3)



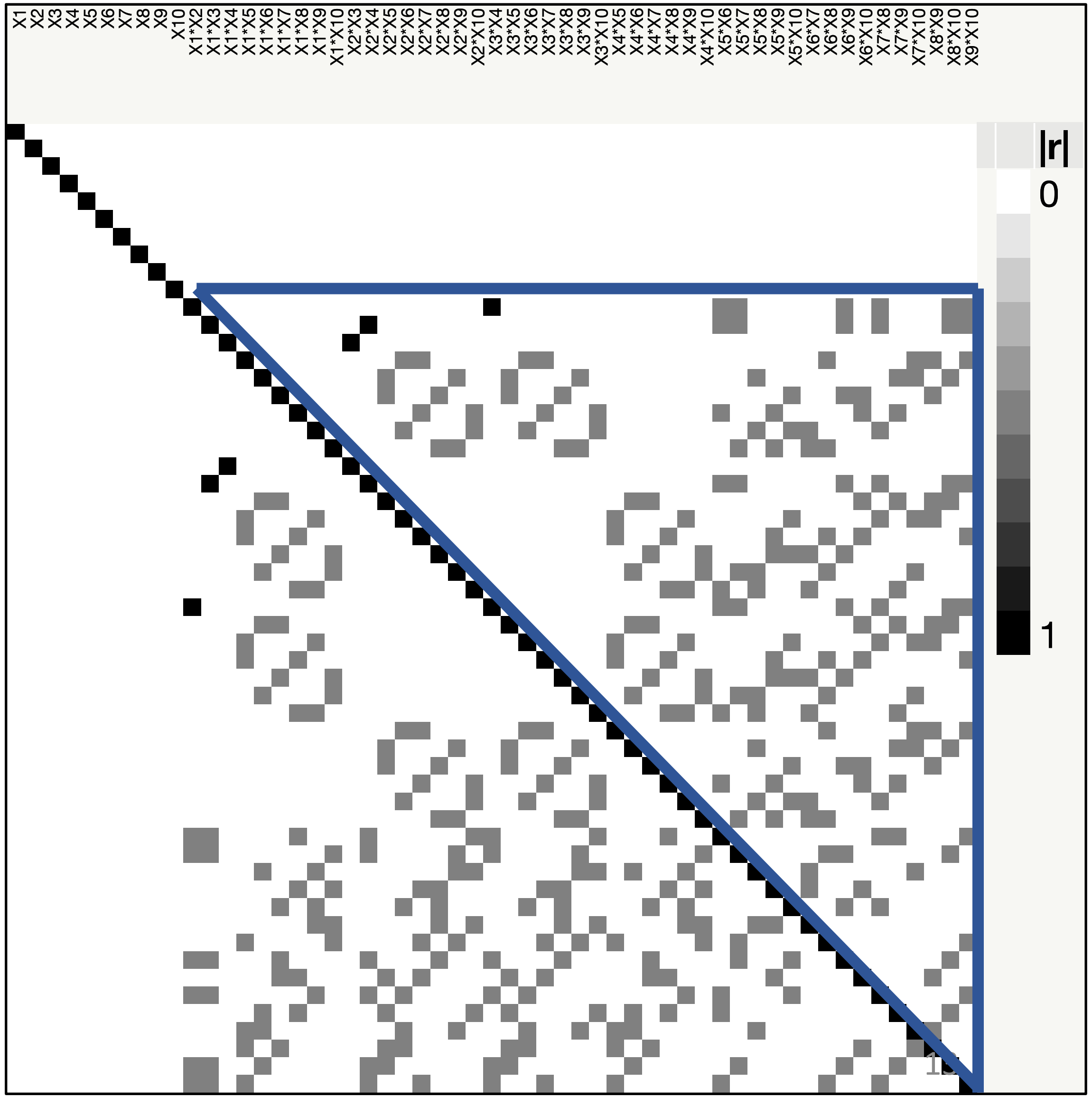
How can we summarize the correlation between pairs of two-factor interactions in strength-3 designs?

The B_4 value equals the sum of squared correlations between pairs of interactions divided by three (Tang & Deng, 1999).

- $B_4 = 1(1)^2 + 62(0.5)^2 = 16.5$

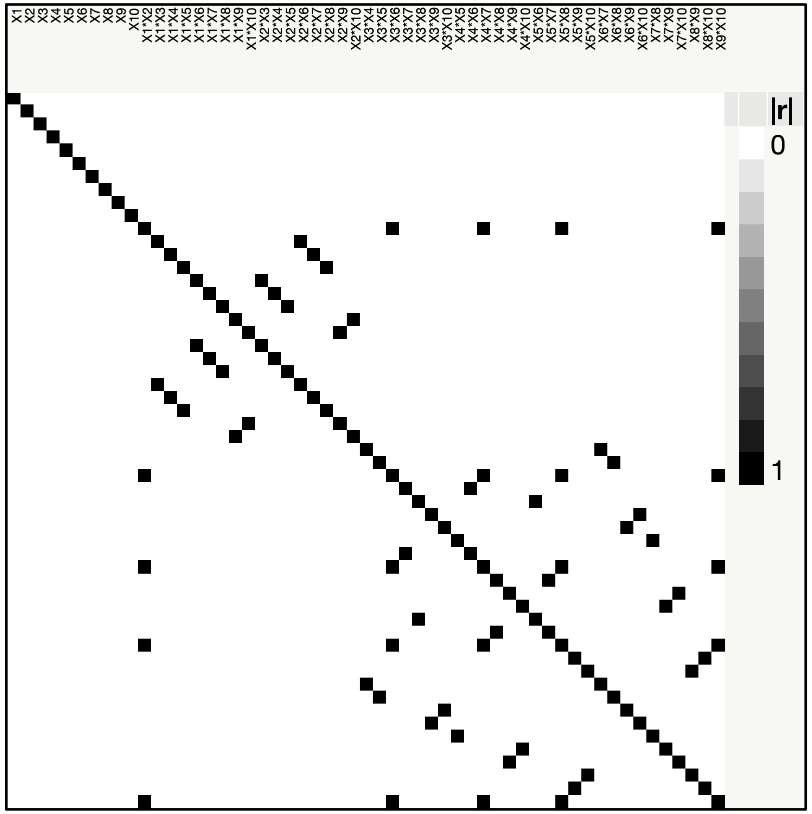
Ideally, the B_4 value of a strength-3 design is small.

Nonregular strength-3 design



Example (cont.):

Regular resolution-IV design

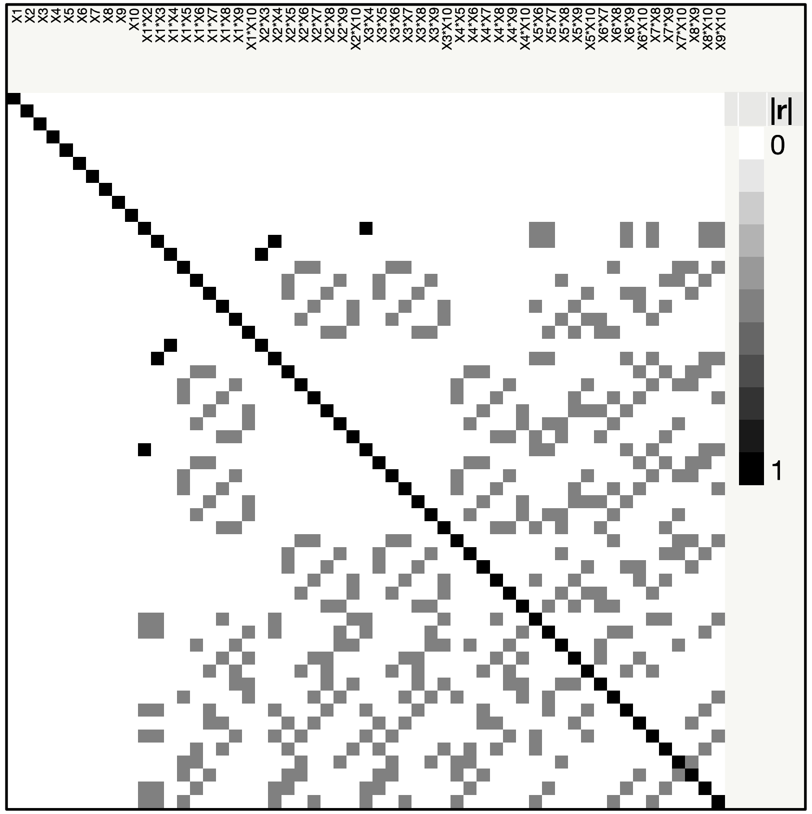


$F_4(1, 0.5) = (10, 0)$

30 pairs with an abs. correlation of 1.

$\boxed{B_4 = 10}$

Nonregular strength-3 design



$F_4(1, 0.5) = (1, 62)$

3 pairs with an abs correlation of 1.

$B_4 = 16.5$

Outline

1. Tuberculosis inhibition experiment
2. Criteria to evaluate designs
3. Construction method
4. Results and discussion

Construction by example

<u>Goal</u>: Construct a strength-3 design with 32 runs and 9 factors.

Step 1. Consider two good equally-sized strength-3 designs with 16 runs and 8 factors. Call them $\boldsymbol{D_u}$ and $\boldsymbol{D_l}$.

Minimum aberration 2_{IV}^{8-4} design

$$\boldsymbol{D_u}, \boldsymbol{D_l} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Vazquez et al. (2019)

Construction by example

Goal: Construct a strength-3 design with 32 runs and 9 factors.

Step 1. Consider two good equally-sized strength-3 designs with 16 runs and 8 factors. Call them $\boldsymbol{D_u}$ and $\boldsymbol{D_l}$.

Step 2. Construct the 32-run 9-factor concatenated design *D.*

$$D = \begin{array}{cc} \left[\begin{array}{c|cccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right] & \begin{array}{c} \boldsymbol{D_u} \\ \\ \\ \boldsymbol{D_l} \end{array} \end{array}$$

Vazquez et al. (2019)

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D*.

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns
  (2) swap columns

Example:
$F_4(1, 0.5) = (14, 0)$

$$D = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]
\begin{array}{l} \boldsymbol{D_u} \\ \\ \boldsymbol{D_l} \end{array}$$

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D*.

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns
  (2) swap columns

Example:
$F_4(1, 0.5) = (7, 0)$

$$
D = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1
\end{array}\right]
$$

Foldover column 8

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D.*

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns
  (2) swap columns

Example:
$F_4(1, 0.5) = (6, 0)$

$$
D = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1
\end{array}\right]
$$

Foldover column 9

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D*.

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns
  (2) swap columns

Example:
$F_4(1, 0.5) = (2, 16)$

Swap columns 6 and 7

$$D = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1
\end{array}\right]$$

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D*.

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns
  (2) swap columns

Example:
$F_4(1, 0.5) = (0, 24)$

$$D = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1
\end{array}\right]$$

Swap columns 7 and 8

Construction by example

Step 3. Use the Column-Change Variable Neighborhood Search algorithm (Vazquez et al., 2019) to sequentially minimize the F_4 vector or minimize the B_4 value of *D.*

- Two moves in $\boldsymbol{D_l}$: (1) foldover columns

  (2) swap columns

Example:

$F_4(1, 0.5) = (0, 24)$

$$D^* = \left[\begin{array}{ccccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1
\end{array}\right]$$

Outline

1. Tuberculosis inhibition experiment
2. Criteria to evaluate designs
3. Construction method
4. Results and discussion

Alternative 14-factor designs for the tuberculosis inhibition experiment

First entry of F_4 vector

Parent Designs (D_u and D_l)	Run Size	Maximum Absolute Correlation Among Interactions	# Pairs of Interactions Involved	B_4 value
Good 40-run strength-3 designs (Schoen et al., 2010).	80	0.60	3	17.56
		0.40	48	19.16
Good 48-run strength-3 designs (Schoen and Mee, 2012).	96	0.34	63	8.33
		0.34	24	10.06
56-run strength-3 designs obtained by dropping columns from the folded-over 28-run Plackett-Burman design…	112	0.43	3	11.00
		0.29	129	11.86

Alternative 14-factor designs for the tuberculosis inhibition experiment

First entry of F_4 vector

Parent Designs	Run Size	Maximum Absolute Correlation Among Interactions	# Pairs of Interactions Involved	B_4 value
Good 48-run strength-3 designs (Schoen and Mee, 2012).	96	0.34	63	8.33
		0.34	24	10.06
Benchmark strength-3 design				
Vazquez and Xu (2019)	96	0.34	639	23.67

Simulation study

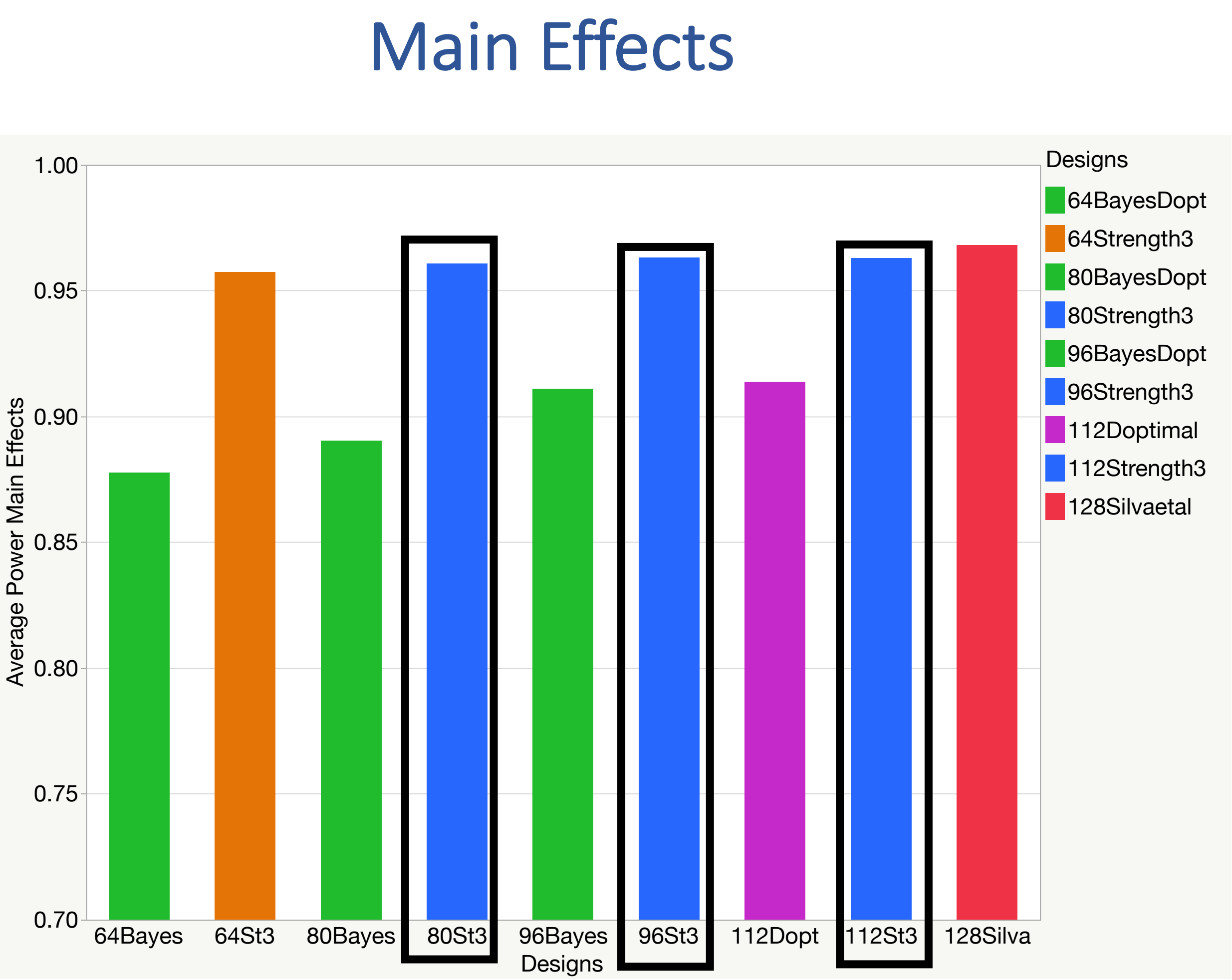
- Using simulations, we evaluate the performance of our 14-factor designs. We restrict our attention to those with best F_4 vector.

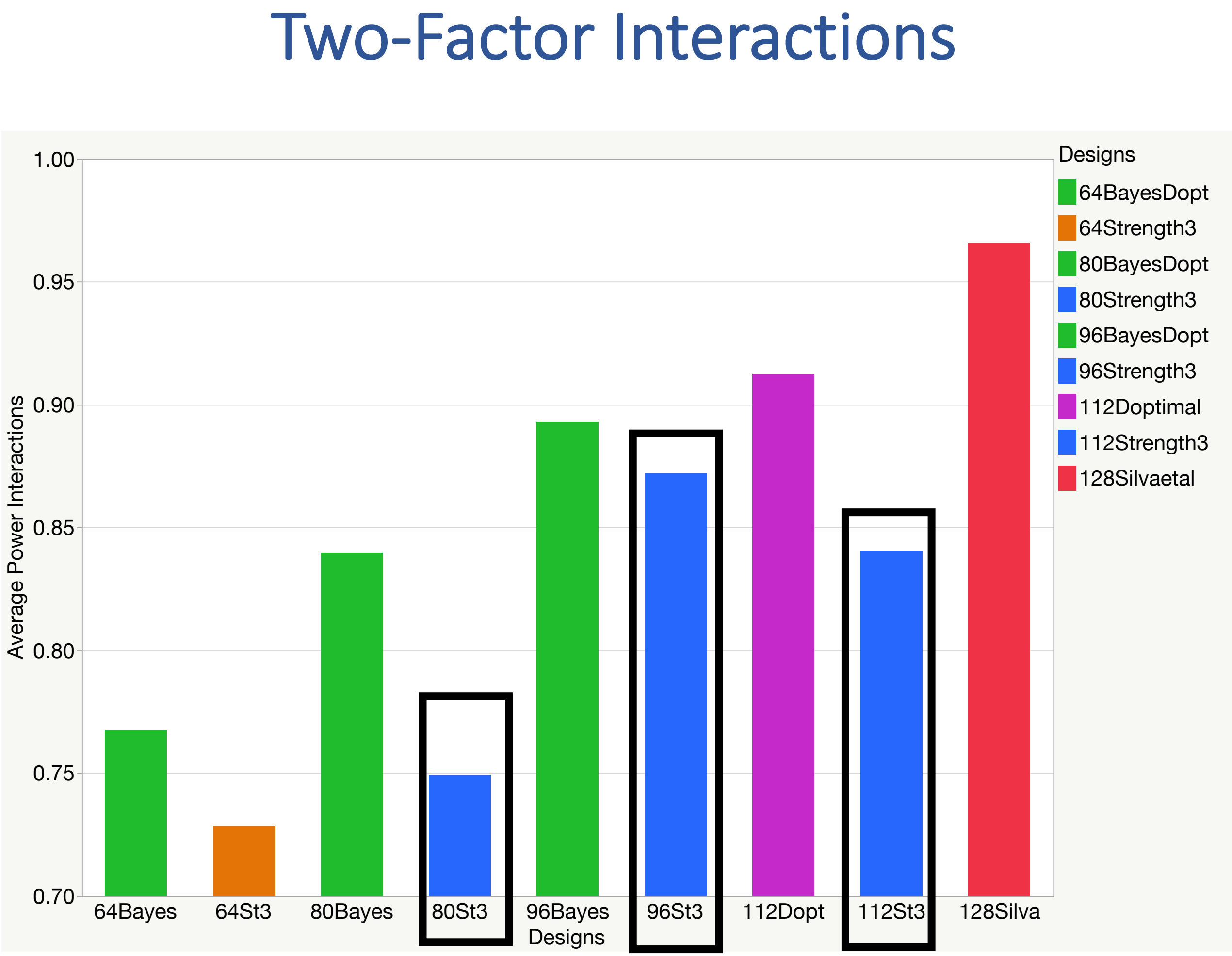
- We also consider two-level benchmark designs in the literature:
  - Strength-3 design with 64 runs (Xu & Wong, 2007)
  - Bayesian D-optimal designs with 64, 80, 96 and 112 runs (DuMouchel & Jones, 1994).
  - D-optimal design with 112 runs (Atkinson et al., 2007).
  - Strength-4 design with 128 runs (Silva et al. , 2016).

Simulation protocol

1. We randomly selected 8 main effects and 6 two-factor interactions (satisfying weak effect heredity) as active.
2. We obtained coefficients for the active effects by adding a 1 to an exponentially distributed random number. A ‘$+$’ or ‘$-$’ was randomly assigned.
3. The coefficients for the inactive effects were drawn from $N(0, 0.25^2)$.
4. We simulated response vectors with residuals following $N(0, 1)$.
5. We used the Dantzig selector (Candes & Tao, 2007) to identify the active effects.

1,000 simulations





Power: Proportion of active effects that are successfully detected.

(Mee, Schoen & Edwards, 2017; Eendebak & Schoen, 2017; Vazquez & Xu, 2019)

Discussion

- For active effects larger than 2, our simulations show that all 80-, 96- and 112-run designs have powers close to 1.

- Using our methodology, we generated a new collection of attractive two-level strength-3 designs with 80, 96 and 112 runs and up to 29 factors.

- Constructing good two-level strength-3 designs with 72, 88, 104 and 120 runs is an interesting topic for future research.

Vazquez, A. R., Schoen, E. D., and Goos, P. (2021). Two-level orthogonal screening designs with 80, 96 and 112 runs, and up to 29 factors. *Journal of Quality Technology*. Published online.