Constructing large OMARS designs by concatenating definitive screening designs

Alan R. Vazquez

Department of Industrial Engineering

University of Arkansas

alanv@uark.edu

mODa 13

9-14 July 2023

UNIVERSITY OF ARKANSAS

Outline

1. Motivating example
2. Construction method for orthogonal minimally aliased response surface (OMARS) designs
3. Numerical comparisons
4. Conclusions

Motivating Example

- Develop a method for extracting pesticides in potato.
- 8 factors under study at 3 levels.
- No more than 40 runs.

		Levels		
Factors		-1	Nominal (0)	1
A	Agitation Time (min)	20	30	40
B	Shaking Time 1 (min)	2	5	8
C	Centrifuge 1 Temperature (ºC)	16	20	24
D	Centrifuge 1 Speed (rpm)	6000	8000	10000
E	Centrifuge 1 Time (min)	3	5	7
F	Shaking Time 2 (min)	2	5	8
G	Centrifuge 2 Temperature (ºC)	16	20	24
H	Centrifuge 2 Time (min)	3	5	7

Joint FAO/IAEA Centre
Nuclear Techniques in Food and Agriculture

Motivating Example

- Develop a method for extracting pesticides in potato.
- 8 factors under study at 3 levels.
- No more than 40 runs.

Design Problem: Construct an efficient experimental design.

Model of Interest

- Full quadratic model in 8 factors.

$$
\begin{aligned}
y = {} & \beta_0 + \beta_1 A + \beta_2 B + \cdots + \beta_8 H \\
& + \beta_{12} AB + \beta_{13} AC + \cdots + \beta_{78} GH \\
& + \beta_{11} A^2 + \beta_{22} B^2 + \cdots + \beta_{88} H^2 \\
& + \epsilon
\end{aligned}
$$

- 1 Intercept
- 8 linear effects
- 28 interactions
- 8 quadratic effects

Total: 45 terms.

- However, the number of effects is larger than the number of runs available.
- Therefore, model-based optimal designs (Goos and Jones, 2011) cannot be used.

Screening Designs

Screening designs allow us to identify the active effects of many factors using an economical number of runs.

To use these designs, we assume that only a few effects are active.

We concentrate on three-level orthogonal screening designs because:

1. They provide linear effects that are not correlated with each other.
2. They allow the study of interactions and quadratic effects.

Available Three-Level Orthogonal Designs

Design	***Number of Runs***										
	17	**20**	**24**	**26**	**27**	**28**	**30**	**32**	**33**	**36**	**40**
Definitive Screening Design (Jones & Nachtsheim, 2011; Xiao et al., 2012)											
Fold-over of Weighing Matrix (Georgiou et al., 2014)											
Orthogonal Array (Cheng & Wu, 2001; Xu et al., 2004)											
Our Proposed Design											

Available Three-Level Orthogonal Designs

Design	***Number of Runs***										
	17	**20**	**24**	**26**	**27**	**28**	**30**	**32**	**33**	**36**	**40**
Definitive Screening Design (Jones & Nachtsheim, 2011; Xiao et al., 2012)											
Fold-over of Weighing Matrix (Georgiou et al., 2014)											
~~Orthogonal Array (Cheng & Wu, 2001; Xu et al., 2004)~~					X					X	
Our Proposed Design											
OMARS designs (Núñez-Arez & Goos, 2020, 2022; Hameed et al., 2023)											

Almost all designs are OMARS designs!

Orthogonal Minimally Aliased Response Surface (OMARS) Designs

OMARS designs are orthogonal designs in which:

- The linear effects are uncorrelated with interactions and quadratic effects.

They are attractive in terms of one or more statistical criteria such as projection and estimation efficiencies (Sun 1999; Lin & Nachtsheim, 2000).

Standard OMARS designs have 3 levels per factor, but extensions exist that accommodate two-level or blocking factors (Núñez-Ares et al., 2023).

(Núñez-Ares & Goos, 2018)

Research question

OMARS designs are currently constructed using an enumeration algorithm (Núnez-Ares & Goos, 2020) that is computationally expensive for large numbers of factors.

In this talk, we introduce an effective method for constructing good standard OMARS designs with large number of quantitative factors.

Outline

1. Motivating example
2. Construction method for orthogonal minimally aliased response surface (OMARS) designs
3. Numerical comparisons
4. Conclusions

Construction by Example

Goal: Construct an 8-factor OMARS design with 33 runs.

Step 1. Consider an 8-factor definitive screening design with 17 runs.

- It is constructed by folding over a conference matrix (Xiao et al., 2012; Schoen et al., 2022).

A	B	C	D	E	F	G	H
0	1	1	1	1	1	1	1
0	-1	-1	-1	-1	-1	-1	-1
1	0	1	1	-1	1	-1	-1
-1	0	-1	-1	1	-1	1	1
1	-1	0	1	1	-1	1	-1
-1	1	0	-1	-1	1	-1	1
1	-1	-1	0	1	1	-1	1
-1	1	1	0	-1	-1	1	-1
1	1	-1	-1	0	1	1	-1
-1	-1	1	1	0	-1	-1	1
1	-1	1	-1	-1	0	1	1
-1	1	-1	1	1	0	-1	-1
1	1	-1	1	-1	-1	0	1
-1	-1	1	-1	1	1	0	-1
1	1	1	-1	1	-1	-1	0
-1	-1	-1	1	-1	1	1	0
0	0	0	0	0	0	0	0

Construction by Example

Step 2. Concatenate two copies of the 8-factor DSD without the center run.

Step 3. Consider column permutations and fold-overs of columns in the lower design to minimize

Sum of squared correlations between:

- Quadratic effect and interaction columns
- Pairs of interaction columns

of D.

$$D = \begin{array}{r} \text{Upper} \\ \\ \\ \\ \\ \\ \\ \\ \text{Lower} \end{array} \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0
\end{array}\right]$$

Construction by Example

Step 2. Concatenate two copies of the 8-factor DSD without the center run.

Step 3. Consider column permutations and fold-overs of columns in the lower design to minimize

Sum of squared correlations between:

- Quadratic effect and interaction columns
- Pairs of interaction columns

of D.

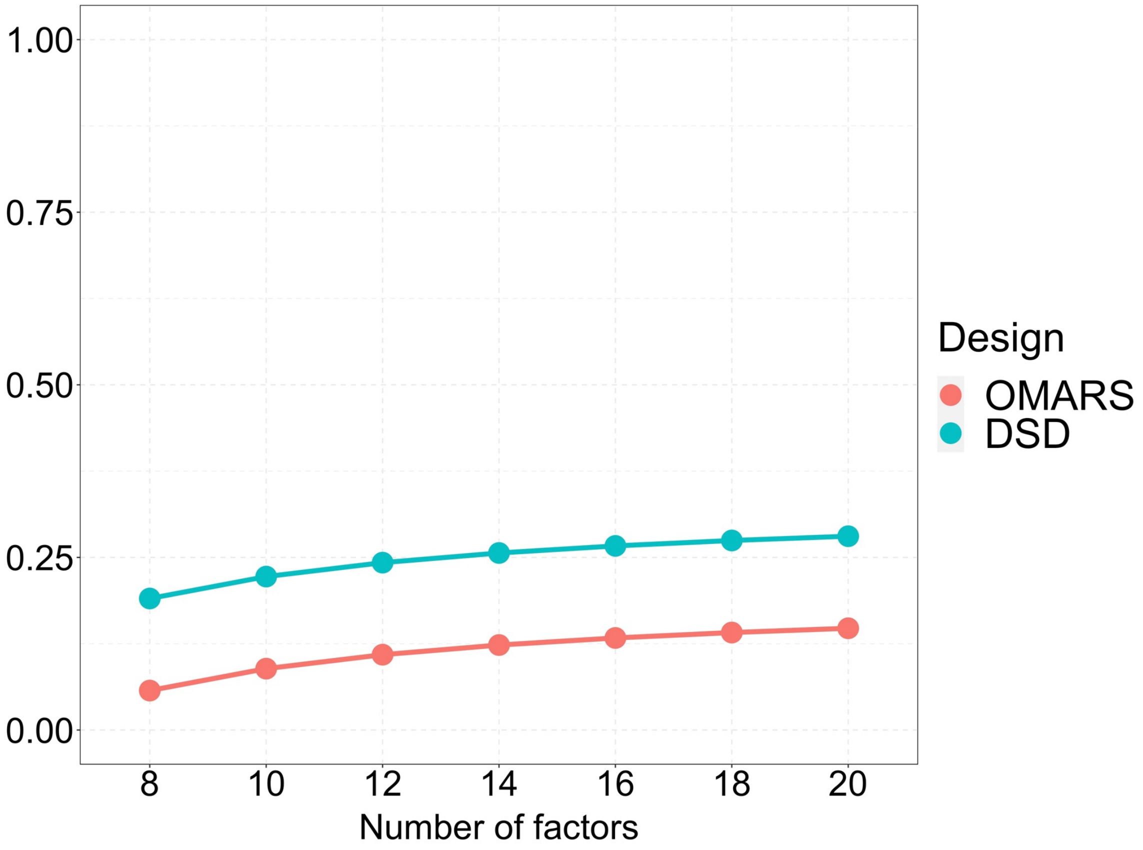
Properties of OMARS designs with an even number of factors *m*

Properties of OMARS designs

The correlation between two quadratic effect columns **does not** depend on column changes in the lower design.

For our *m*-factor OMARS design, the absolute correlation between two quadratic effect columns is:

$$\frac{m-6}{5(m-1)}.$$



Correlation between a quadratic effect and an interaction column

Type 1: Not sharing a factor.

Example: A^2 and BC

For our m-factor OMARS design, this correlation is 0.

Type 2: Sharing a factor.

Example: A^2 and AB

For our m-factor OMARS design, the absolute correlation is:

$$0 \quad \text{or} \quad \sqrt{\frac{4m+1}{5(m-1)(m-2)}}.$$

Correlation between a quadratic effect and an interaction column

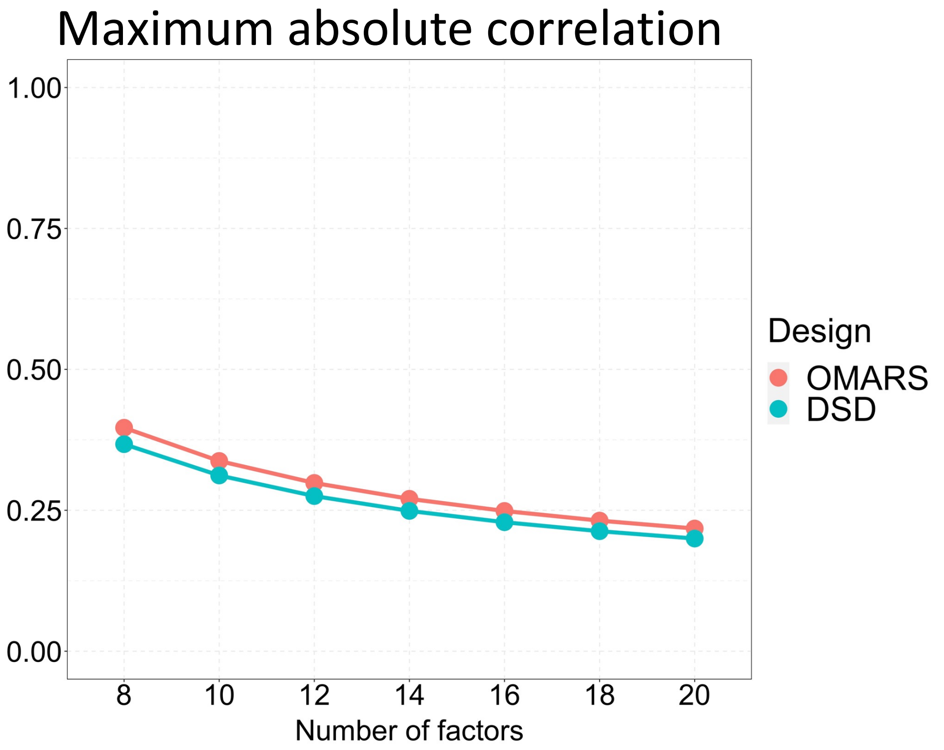
Type 1: Not sharing a factor.

Example: A^2 and BC

For our *m*-factor OMARS design, this correlation is 0.

Type 2: Sharing a factor.

Example: A^2 and AB



Correlation between two interaction columns

Type 1: Sharing a factor.

Example: AB and AC.

Type 2: Not sharing a factor.

Example: AB and CD.

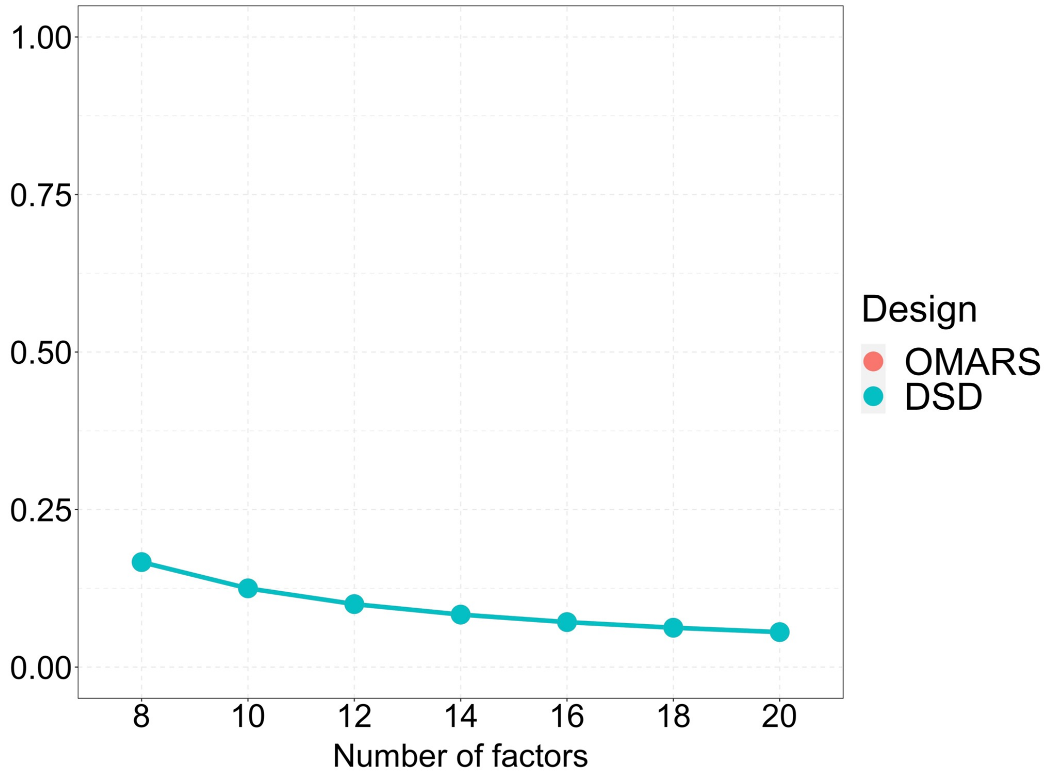
Correlation between two interaction columns

Type 1: Sharing a factor.

Example: AB and AC.

For our m-factor OMARS design, this correlation is:

$$0 \quad \text{or} \quad \frac{1}{m-2}.$$



Type 2: Not sharing a factor.

Example: AB and CD.

Theorem 1. **If *m* is a multiple of 4**, the absolute correlation between two interaction columns involving four factors in our m-factor OMARS design can be

$$\frac{m - 2\lambda}{m-2} \quad \text{for } \lambda = 2, 3, \ldots, m/2.$$

Example: For the 8-factor designs, we have

- DSD: 0.667 and 0.
- OMARS: 0.667, 0.333, and 0.

Type 2: Not sharing a factor.

Example: AB and CD.

Theorem 2. **If *m* is 2 more than a multiple of 4**, the absolute correlation between two interaction columns involving four factors in our m-factor OMARS design can be

$$\frac{4\lambda}{m-2} \quad \text{or} \quad \frac{m-4(\lambda+1)}{m-2} \quad \text{for } \lambda = 0, 1, \ldots, (m-6)/4.$$

Construction by Example

Step 2. Concatenate two copies of the 8-factor DSD without the center run.

Step 3. Consider column permutations and fold-overs of columns in the lower design to <u>minimize</u>

Sum of squared correlations between:

- Quadratic effect and interaction columns
- Pairs of interaction columns

of D.

We are back!

$$D = \begin{array}{c} \text{Upper} \\ \\ \\ \text{Lower} \end{array} \left[\begin{array}{rrrrrrrr}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0
\end{array}\right]$$

Motivating problem

Evaluating all possible concatenated designs D would require
$8! \text{ x } 2^8$ = 10,321,920 evaluations.

$$D = \begin{array}{ll}
\text{Upper} & \left[\begin{array}{rrrrrrrr}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\ \hline
\end{array}\right. \\
\text{Lower} & \left.\begin{array}{rrrrrrrr}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0
\end{array}\right]
\end{array}$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value $= 74.57$

$$D = \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0
\end{array}\right]$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value $= 46.74$

Fold-over column 8

$$D = \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & -1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0
\end{array}\right]$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value $= 42.81$

Swap columns 8 and 9

$$
D = \left[\begin{array}{rrrrrrrr}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & -1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & 1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 0 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & -1 & 0 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & -1 & 1 & -1 & 0 & -1 \\
-1 & -1 & -1 & 1 & -1 & 1 & 0 & 1
\end{array}\right]
$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value $= 38.01$

Swap columns 1 and 4

$$
D = \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
-1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & 1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 \\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
0 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & -1 & 1 & 0 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 0 & -1 & -1 & -1 \\
-1 & -1 & 1 & 1 & -1 & 0 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 0 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & -1 & 0 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & 0 \\
-1 & 1 & 1 & 1 & 1 & -1 & 0 & -1 \\
1 & -1 & -1 & -1 & -1 & 1 & 0 & 1
\end{array}\right]
$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value $= 37.57$

$$D = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
-1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & 1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & -1 & 1 \\
-1 & -1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & -1 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 \\
-1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 \\
1 & 1 & -1 & 1 & 0 & -1 & 1 & 1 \\
-1 & -1 & 1 & -1 & 0 & 1 & -1 & -1 \\
0 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\
0 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \\
-1 & 1 & 1 & 1 & 1 & -1 & 0 & -1 \\
1 & -1 & -1 & -1 & -1 & 1 & 0 & 1
\end{bmatrix}$$

Swap columns 1 and 6

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value = 36.21

Swap columns 1 and 3

$$D = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
-1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & 1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & -1 & 1 \\
0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & 1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & 1 & 1 & 0 & -1 & 1 & 1 \\
1 & -1 & -1 & -1 & 0 & 1 & -1 & -1 \\
1 & -1 & 0 & 1 & -1 & -1 & -1 & 1 \\
-1 & 1 & 0 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \\
1 & 1 & -1 & 1 & 1 & -1 & 0 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 0 & 1
\end{bmatrix}$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value = 32.72

Fold-over columns 4 and 5

$$D = \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 \\
-1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 \\
1 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\
0 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 \\
-1 & 1 & 1 & -1 & 0 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 \\
1 & -1 & 0 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & 0 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & 0 \\
1 & 1 & -1 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & 1 & 1 & 1 & 1 & 0 & 1
\end{array}\right]$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Objective Value = 32.28

Swap columns 1 and 5

$$
D = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
-1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 0 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 0 \\
-1 & 1 & -1 & -1 & 1 & -1 & 0 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & 0 & 1
\end{bmatrix}
$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Locally optimal design

$$
D = \left[\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
-1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 0 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 0 \\
-1 & 1 & -1 & -1 & 1 & -1 & 0 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & 0 & 1
\end{array}\right]
$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Step 4. Add a row of zeros.

$$
D = \left(\begin{array}{rrrrrrrr}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
\hline
-1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 0 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 0 \\
-1 & 1 & -1 & -1 & 1 & -1 & 0 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
$$

Algorithmic Approach

Column-Change Variable Neighborhood Search (CC-VNS) algorithm (Vazquez et al., 2018)

Two *moves:*

(1) fold-over columns.
(2) swap two columns.

Output: 8-factor OMARS design with 33 runs.

$$D = \begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
-1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\
1 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 0 & -1 & 1 & -1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & -1 & 0 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 0 \\
-1 & 1 & -1 & -1 & 1 & -1 & 0 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Outline

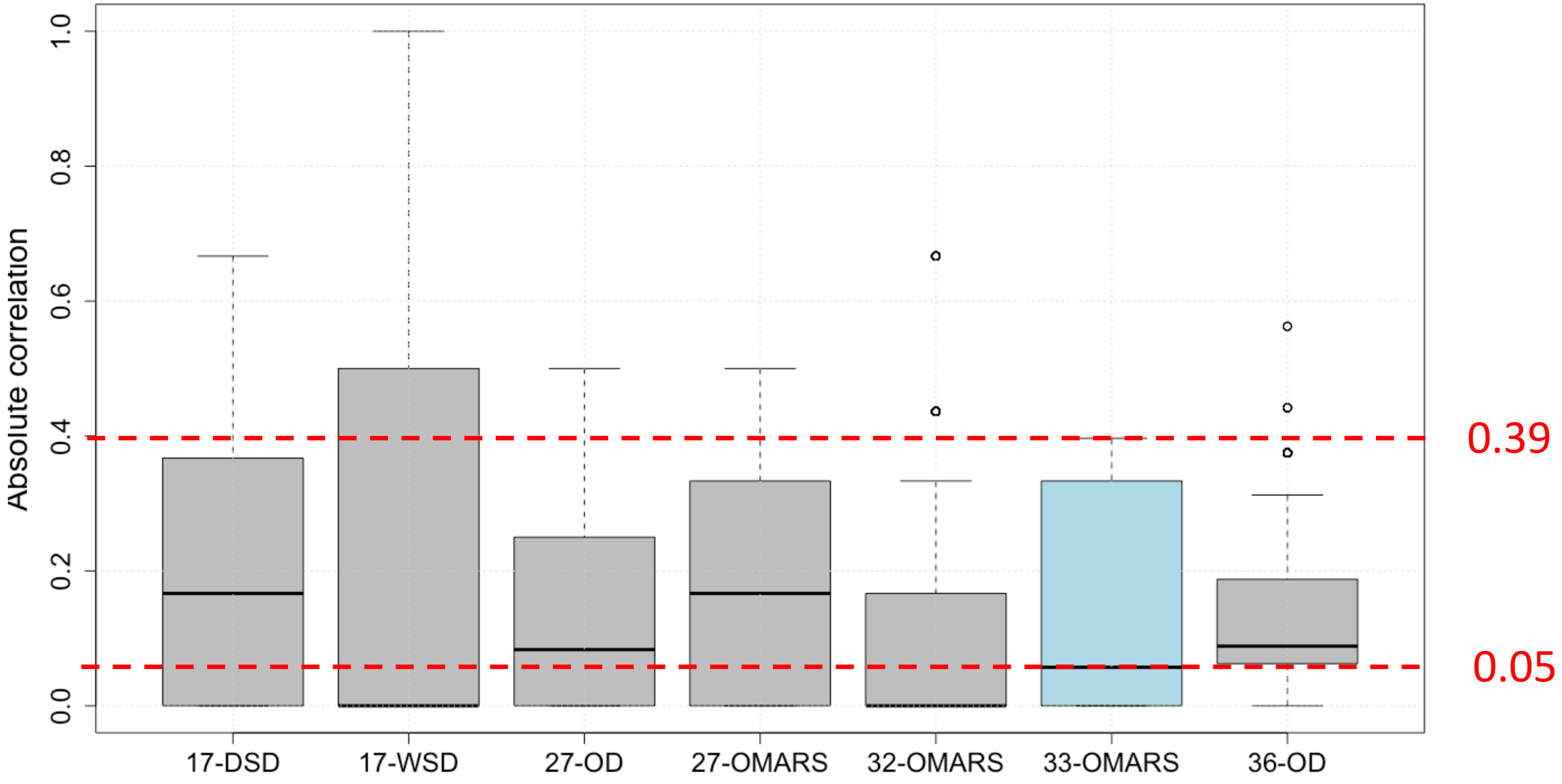
1. Motivating example
2. Construction method for orthogonal minimally aliased response surface (OMARS) designs
3. Numerical comparisons
4. Conclusions

Comparison with other 8-factor designs

We compare our 8-factor 33-run OMARS designs with other three-level orthogonal designs:

- **17-DSD**: 17-run Definitive Screening Design (Jones & Nachtsheim, 2011).
- **17-WSD**: 17-run design obtained by folding over a weighing matrix (Georgiou et al., 2014).
- **27-OD**: 27-run nonregular design (Xu et al., 2004).
- **27-OMARS**: 27-run OMARS design (Hameed et al., 2023).
- **32-OMARS**: 32-run OMARS design (Hameed et al., 2023).
- **36-OD**: 36-run nonregular design (Cheng & Wu, 2001).

Absolute correlation between pairs of second-order effect columns (quadratic effects and interaction effects).



Outline

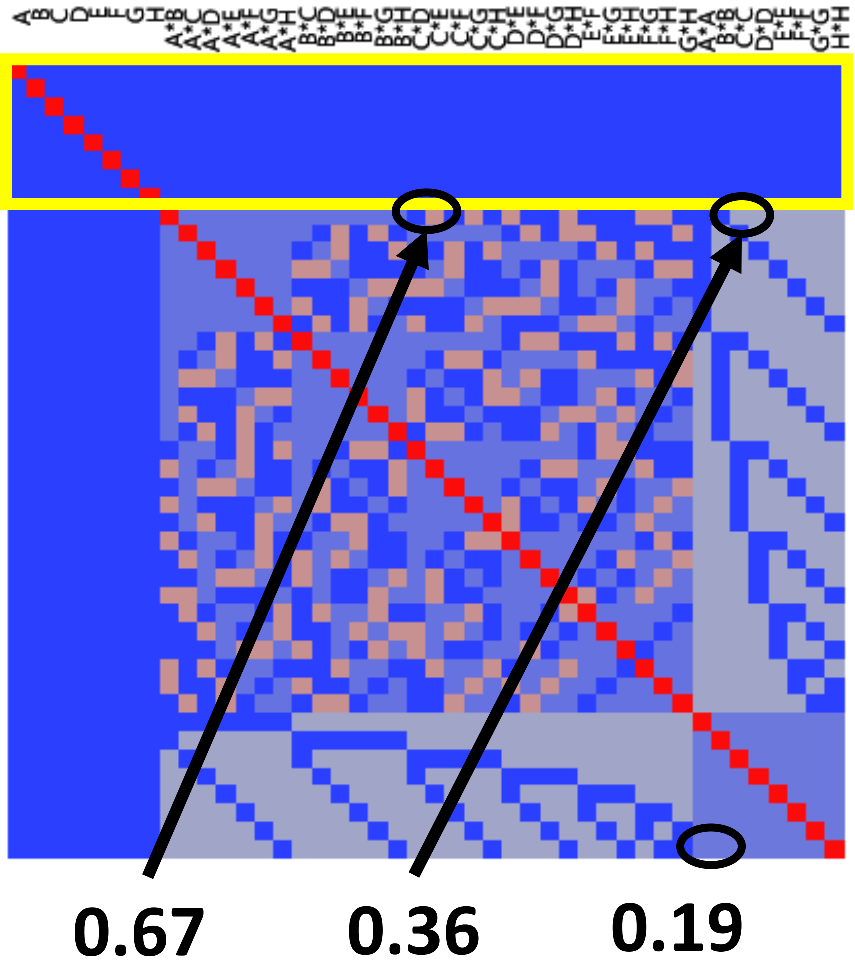
1. Motivating example
2. Construction method for orthogonal minimally aliased response surface (OMARS) designs
3. Numerical comparisons
4. Conclusions

Conclusions

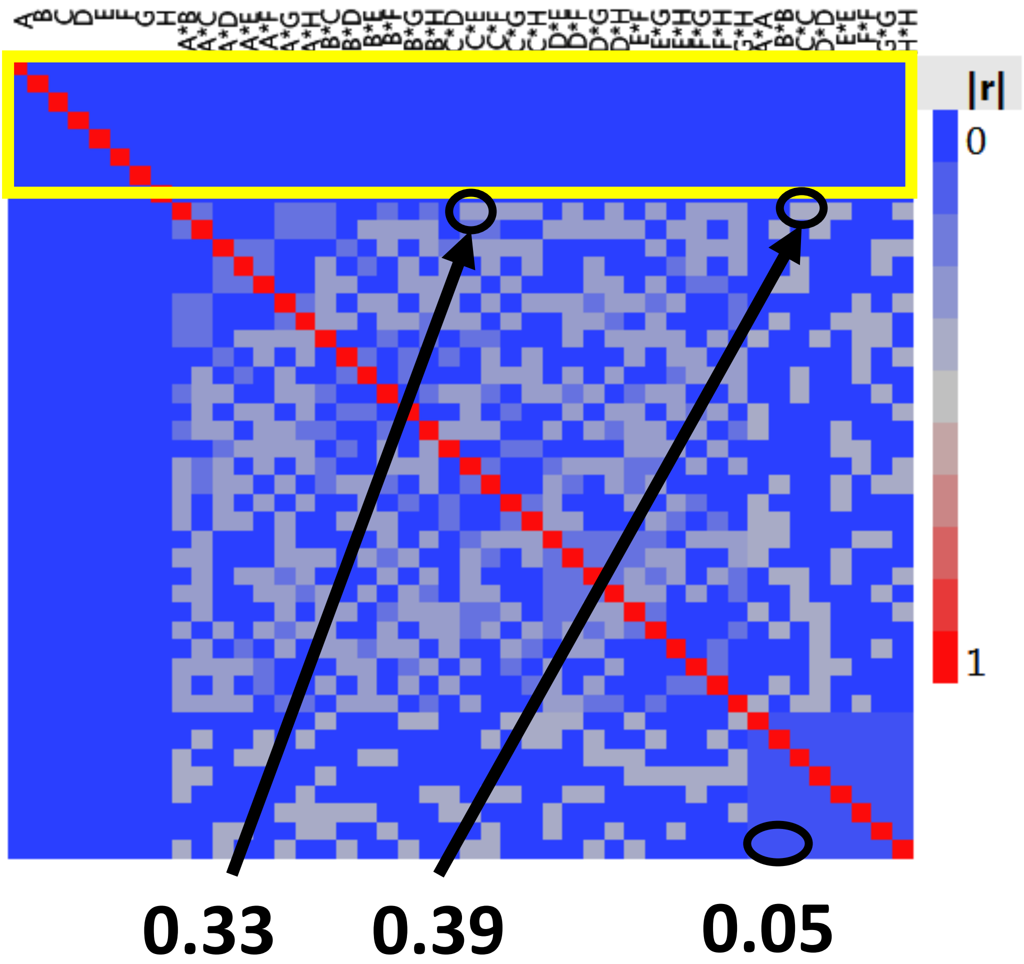
- Our 33-run 8-factor OMARS design is competitive with the benchmark designs.
- Our construction method can generate attractive OMARS designs with an even number of factors. For odd numbers of factors, drop one column from our designs.
- In the end, a variant of our 33-run 8-factor OMARS design was used in the extraction experiment (Maestroni, Vazquez, Goos, et al., 2018). It collected observations on 24 responses.
- The data analysis showed that some factors have significant interactions and quadratic effects on several responses.

Email: alanv@uark.edu

Appendix



Definitive Screening Design
0.67
0.36
0.19
17 observations



Concatenated
Definitive Screening Design
|r|
0
1
0.33
0.39
0.05
33 observations

References

- Cheng, S., & Wu, C. F. J. (2001). Factor Screening and Response Surface Exploration. *Statistica Sinica*, *11*, 553–604.
- Goos, P. and Jones, B. (2011). *Optimal Design of Experiments: A Case Study Approach*. Wiley.
- Vazquez, A. R., Goos, P., and Schoen, E. D. (2019). Constructing two-level designs by concatenation of strength-3 orthogonal arrays. *Technometrics*, 61:219–232.
- Maestroni, B. M., Vazquez, A. R., Avossa, V., Goos, P., Cesio, V., Heinzen, H., Riener, J., Cannavan, A. (2018). Ruggedness testing of an analytical method for pesticide residues in potato. *Accreditation and Quality Assurance*, 23:303-316.
- Schoen, E. D., Eendebak, P. T., Vazquez, A. R., and Goos, P. (2022). Systematic enumeration of definitive screening designs. *Statistics and Computing*. Published Online.
- Jones, B. & Nachtsheim, C.J. (2011). A Class of Three-Level Designs for Definitive Screening in the Presence of Second-Order Effects. *Journal of Quality Technology*, 43:1, 1-15.
- Xiao, L., Lin, D. K. J. & Bai, F. (2012). Constructing Definitive Screening Designs Using Conference Matrices. *Journal of Quality Technology*, 44:1, 2-8.
- Hameed, M. S. I., Núñez-Ares, J. & Goos, P. (2023). Analysis of data from orthogonal minimally aliased response surface designs. *Journal of Quality Technology*, 55:3, 366-384.
- Núñez-Ares, J. & Goos, P. (2020). Enumeration and Multicriteria Selection of Orthogonal Minimally Aliased Response Surface Designs. *Technometrics*, 62:1, 21-36.
- Núñez-Ares, J. & Goos, P. (2023). Blocking OMARS designs and definitive screening designs. *Journal of Quality Technology.* Published online.
- Xu, H., Cheng, S.-W. & Wu, C.F.J (2004). Optimal Projective Three-Level Designs for Factor Screening and Interaction Detection. *Technometrics*, 46:3, 280-292.
- Georgiou, S. D., Stylianou, S. & Aggarwal, M. (2014). Efficient three-level screening designs using weighing matrices. *Statistics*, 48:4, 815-833.