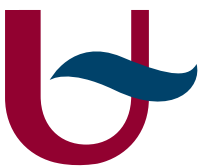
Fractional Factorial Designs by Combining Two-Level Designs

Alan Vázquez, Peter Goos & Eric Schoen

University of Antwerp

alan.vazquezalcocer@uantwerpen.be

BSS Meeting 2015



Universiteit Antwerpen

Outline

1. Introduction

2. Evaluating Experimental Designs

3. Construction Method

4. Results and Conclusions

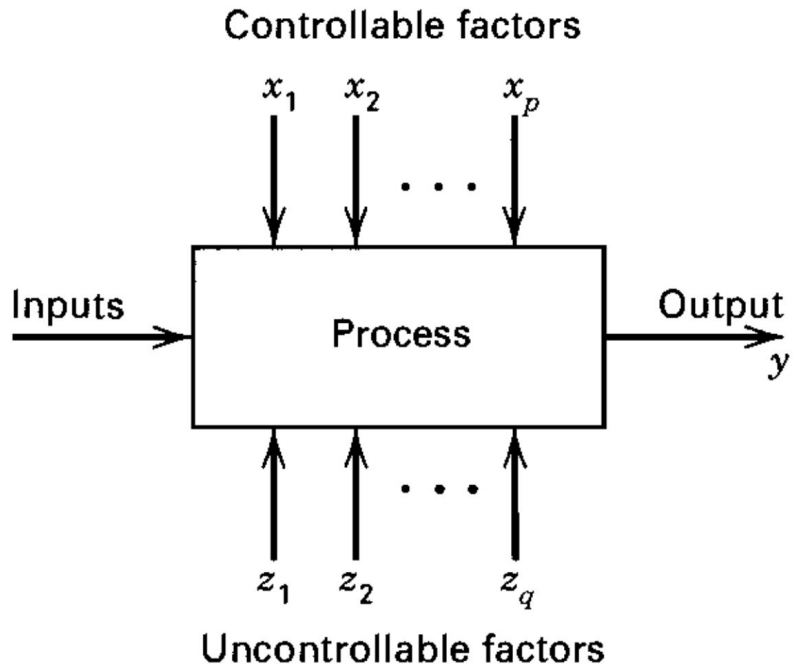
Outline

Introduction: Experimental design

A structured plan for performing a series of tests of a system, a process or a product

- Model: $y = f(x_1, \ldots, x_p) + \epsilon$
- Interest is in finding economical experimental plans

$f(x_1, \ldots, x_p)$ includes main effects, two-factor interactions, quadratic effects, cubic effects, etc.



Neural Network Training Experiment

- Investigate the architecture and training of a neural network to predict short-term load requirements for an Ohio electric utility (Mee, 2009)

Neural Network Training Experiment

- Investigate the architecture and training of a neural network to predict short-term load requirements for an Ohio electric utility (Mee, 2009)
- Ten factors under study each at two levels

		Levels	
Factors		-1	1
A	Hidden layers	1	2
B	Transfer function in output layer	Linear	Sigmoid
C	Transfer function in hidden layer	Sigmoid	Sinusoid
D	Backpropagation learning algorithm	Standard	Cumulative
E	Gaussian noise added	No	Yes
F	Stopping rule	RMSE	CD
G	Network	Feedforward	Recurrent
H	Years of training data	2	4
J	Time of peak	Winter	Summer
K	Industrial load	Low	High

Neural Network Training Experiment

- Investigate the architecture and training of a neural network to predict short-term load requirements for an Ohio electric utility (Mee, 2009)
- Ten factors under study each at two levels
- Budget allows for 64 observations

		Levels	
Factors		-1	1
A	Hidden layers	1	2
B	Transfer function in output layer	Linear	Sigmoid
C	Transfer function in hidden layer	Sigmoid	Sinusoid
D	Backpropagation learning algorithm	Standard	Cumulative
E	Gaussian noise added	No	Yes
F	Stopping rule	RMSE	CD
G	Network	Feedforward	Recurrent
H	Years of training data	2	4
J	Time of peak	Winter	Summer
K	Industrial load	Low	High

Neural Network Training Experiment

- Investigate the architecture and training of a neural network to predict short-term load requirements for an Ohio electric utility (Mee, 2009)
- Ten factors under study each at two levels
- Budget allows for 64 observations

Design Problem:

Construct an efficient experimental plan

Neural Network Training Experiment

Our first option for the experimental plan might be to test all the level combinations of the factors: $2^{10} = 1024$ observations!

Neural Network Training Experiment

Our first option for the experimental plan might be to test all the level combinations of the factors: $2^{10} = 1024$ observations!

Prior information:

1. Most of the main effects of the 10 factors will be active
2. Considerable number of active two-factor interactions
3. Three-factor and higher order interactions are negligible

Neural Network Training Experiment

Our first option for the experimental plan might be to test all the level combinations of the factors: $2^{10} = 1024$ observations!

Prior information:

1. Most of the main effects of the 10 factors will be active
2. Considerable number of active two-factor interactions
3. Three-factor and higher order interactions are negligible

How to select a fraction of the 1024 level combinations?

Experimental plans

Full Factorial designs:

A	B	C	AB	AC	BC	ABC
−1	−1	−1	1	1	1	−1
1	−1	−1	−1	−1	1	1
−1	1	−1	−1	1	−1	1
1	1	−1	1	−1	−1	−1
−1	−1	1	1	−1	−1	1
1	−1	1	−1	1	−1	−1
−1	1	1	−1	−1	1	−1
1	1	1	1	1	1	1

3 factors and 8 runs

Experimental plans

Full Factorial designs:

A	B	C	AB	AC	BC	ABC
−1	−1	−1	1	1	1	−1
1	−1	−1	−1	−1	1	1
−1	1	−1	−1	1	−1	1
1	1	−1	1	−1	−1	−1
−1	−1	1	1	−1	−1	1
1	−1	1	−1	1	−1	−1
−1	1	1	−1	−1	1	−1
1	1	1	1	1	1	1

Experimental plans

Regular Fractional Factorial designs:

A	B	C	AB	AC	BC	ABC
1	−1	−1	−1	−1	1	1
−1	1	−1	−1	1	−1	1
−1	−1	1	1	−1	−1	1
1	1	1	1	1	1	1

3 factors and 4 runs

Aliased contrasts:

C = AB, A = CB, B = AC, ABC = I

Outline

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

Three options for the Neural Network Experiment:

- Regular Fractional Factorial design (Chen et al., 1993)
- Quaternary linear codes (Xu & Wong, 2007)
- Concatenating smaller designs (Authors)

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

These designs are *orthogonal arrays of strength three*.

1. Main effects are not correlated with two-factor interactions
2. Pairs of two-factor interactions can be correlated

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

How can we measure the correlation between pairs of two-factor interactions (2FI)?

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

The possible absolute values for the correlations between pairs of 2FI are 1, 0.75, 0.5, 0.25, and 0 (Deng & Tang, 1999).

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

The F_4 vector has entries equal to the frequencies for the possible absolute correlation values between pairs of 2FI, divided by three (Deng & Tang, 1999)

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

Example:

Chen et al. (1993):

$$E(\hat{\beta}_{AB}) = \beta_{AB} + 1\beta_{CG}$$

Xu & Wong (2007):

$$E(\hat{\beta}_{AC}) = \beta_{AC} + 0.5\beta_{GJ}$$

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

The F_4 vector has entries equal to the frequencies for the possible absolute correlation values between pairs of 2FI, divided by three (Deng & Tang, 1999)

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

The A_4 value is proportional to the sum of squared correlations between pairs of two-factor interactions
(Tang & Deng, 1999)

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

The number of estimable two-factor interactions equals the rank(X_2) (Schoen & Mee, 2012)

- X_2 is the matrix containing the two-factor interactions contrast vectors

Designs with 64 runs and 10 factors

Design	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Estimable 2FI
Chen et al. (1993)	(2, 0, 0, 0)	2	39
Xu & Wong (2007)	(0, 0, 8, 0)	2	39
Authors	(0, 0, 0, 32)	2	45

Outline

Construction by Example

Step 1 Concatenate two designs D_u and D_l

16 runs and 8 factors, 2_{IV}^{8-4}

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

32 runs and 8 factors

Construction by Example

Step 1 Concatenate two designs D_u and D_l

16 runs and 8 factors, 2_{IV}^{8-4}

Step 2 Consider column permutations and sign-reverse of columns in D_l to

(a) sequentially minimize the F_4 vector or

(b) minimize the A_4 value

of D

$D =$

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

D_u

D_l

32 runs and 8 factors

Construction by Example

Step 1 Concatenate two designs D_u and D_l

16 runs and 8 factors, 2^{8-4}_{IV}

Step 2 Consider column permutations and sign-reverse of columns in D_l to

(a) sequentially minimize the F_4 vector or

(b) minimize the A_4 value

of D

$\star$ Evaluating all possible combined designs D would require $8! \times 2^8 = 10,321,920$ evaluations

$D =$

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

D_u

D_l

32 runs and 8 factors

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

$F_4(1, 0.5) = (14,\ 0)$

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

$F_4(1, 0.5) = (7,\ 0)$

Sign-reverse column 7

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

$F_4(1, 0.5) = (6,\ 0)$

Sign-reverse column 8

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

$F_4(1, 0.5) = (2,\ 16)$

Swap columns 5 and 6

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

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

Swap columns 6 and 7

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

An algorithmic approach

- Variable Neighborhood Search (VNS)
  — Framework to construct efficient algorithms
- Two *moves*: (1) sign-reverse (2) swap columns

Example:

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

Less than 2% of the total number of evaluations

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

Construction by Example

Step 1 Concatenate two designs D_u and D_l

16 runs and 8 factors, 2_{IV}^{8-4}

Step 2 Consider column permutations and sign-reverse of columns in D_l to

(a) sequentially minimize the F_4 vector or

(b) minimize the A_4 value

of D

Step 3 Add an extra orthogonal column

$D =$

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

32 runs and 9 factors

Outline

Results

- According to Schoen et al. (2010), there are 34 strength-3 OAs with 32 runs and 9 factors.
- We tested all possible combinations of OAs as D_u and D_l and used the proposed methodology.

Efficient experimental plan with 64 runs and 10 factors

$\{D_u, D_l\}$	$F_4(1, 0.75, 0.5, 0.25)$	A_4	# Est. 2FI
$\{27, 34\}$	(0, 0, 0, 32)	2	45

Conclusions

This methodology can be used to construct nonregular designs with 64, 80, 96, 112, and 128 runs, and up to 33 factors.

The VNS algorithm provides a fast and reliable alternative to improve the combined designs.

We can combine other orthogonal arrays such as strength-2 OAs, mixed-level OAs, etc.

See Dragulić et al. (2014) for analysis strategies.

References

Chen, J., Sun, D.X. & Wu, C.F.J. (1999). A catalogue of two-level and three-level fractional factorial designs with small runs. *Internat. Statist. Rev.*, 61, p.p. 131-145.

Deng, L.-Y. & Tang, B. (1999). Generalized resolution and minimum aberration criteria for Plackett-Burman and other nonregular factorial design. *Statistica Sinica*, 9, p.p. 1071-1082.

Dragulić, D., Woods, D.C., Dean, A.M., Lewis, S.M., & Vine, A.-J.E. (2014). Screening strategies in the presence of interactions. *Technometrics* 56(1), p.p. 1-1.

Mee, R. (2009). *A Comprehensive Guide to Factorial Two-Level Experimentation*. Springer.

Schoen, E.D. & Mee, R.W. (2012). Two-level designs of strength-3 and up to 48 runs. *Appl. Statist.* 61. Part 1, p.p. 163-174.

Schoen, E.D., Eendebak, P.T. & Nguyen, M.V.M. (2010). Complete enumeration of pure-level and mixed-level orthogonal arrays. *Journal of Combinatorial Designs*, 18(2), p.p. 123-140.

Tang, B. and Deng, L.-Y. (1999). Minimum G_2-aberration for nonregular fractional factorial designs. *The Annals of Statistics*, 27(6), p.p 1914-1926.