Constructing Optimal Screening Designs for Effective Experimentation using Metaheuristics

Alan R. Vazquez

University of California, Los Angeles

alanrvazquez@stat.ucla.edu

Metaheuristic Optimization, Machine Learning and AI – Virtual Workshop

March 8-12, 2021

Funded by the Research Foundation – Flanders, Belgium

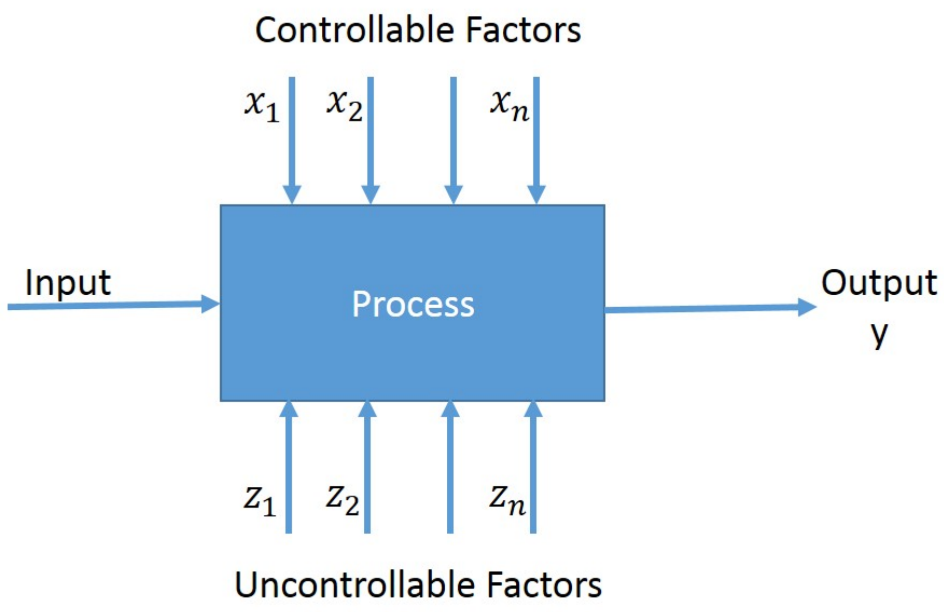
Outline

1. Introduction
2. Q_B criterion for evaluating experimental designs
3. A metaheuristic algorithm for constructing efficient designs
4. Results and Conclusions

Experimental design

Structured plan for performing a series of tests of a system, a process or a product.

- Model: $y = f\left(x_1, \dots, x_p\right) + \varepsilon$.
- Interest is in finding economical experimental designs.



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

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.

Tuberculosis inhibition experiment

The first option for the experiment might be to test all the level combinations of the factors: $2^{14} = 16{,}384 \text{ tests!}$

Prior information:

1. Most of the 14 factors would be active.
2. Considerable number of active two-factor interactions.
3. Negligible three-factor and higher-order interactions.

How to select a fraction of the 16,384 level combinations?

For the experiment, the researchers chose an attractive experimental design with 14 factors and 128 test combinations (runs).

This design allowed the estimation of 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?

Outline

1. Introduction
2. Q_B criterion for evaluating experimental designs
3. A metaheuristic algorithm for constructing efficient designs
4. Results and Conclusions

Evaluating two-level designs

Example: Compare two-level designs with 24 runs and 7 factors.

Design 1.

Folded-over Plackett-Burman design

(Miller & Sitter, 2001)

X1	X2	X3	X4	X5	X6	X7
1	-1	1	1	1	-1	-1
1	-1	-1	-1	1	-1	1
1	-1	1	1	-1	1	1
1	1	1	-1	-1	-1	1
-1	1	-1	1	1	-1	1
-1	1	1	-1	1	1	1
-1	1	1	1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	1	-1	1	1	-1
1	1	-1	-1	-1	1	-1
1	1	-1	1	1	1	-1
-1	-1	-1	1	-1	1	1
-1	1	-1	-1	-1	1	1
-1	1	1	1	-1	1	-1
-1	1	-1	-1	1	-1	-1
-1	-1	-1	1	1	1	-1
1	-1	1	-1	-1	1	-1
1	-1	-1	1	-1	-1	-1
1	-1	-1	-1	1	1	1
1	1	1	1	1	1	1
1	1	-1	1	-1	-1	1
-1	-1	1	1	1	-1	1
-1	-1	1	-1	-1	-1	1
1	1	1	-1	1	-1	-1

Design 2.

Obtained from

http://neilsloane.com/hadamard/

X1	X2	X3	X4	X5	X6	X7
-1	-1	1	1	1	1	1
1	-1	1	-1	-1	-1	-1
-1	1	1	1	1	1	-1
-1	-1	-1	-1	-1	-1	-1
1	1	1	1	-1	1	-1
1	1	-1	1	-1	1	1
1	1	1	1	1	-1	1
1	-1	-1	-1	-1	1	1
-1	1	-1	1	-1	-1	-1
1	-1	-1	1	-1	1	-1
1	-1	-1	1	1	-1	-1
-1	1	-1	-1	-1	-1	1
-1	-1	-1	-1	1	1	1
-1	1	1	-1	-1	1	-1
-1	-1	1	-1	1	-1	-1
1	-1	1	-1	1	1	-1
-1	-1	-1	1	1	1	1
-1	-1	1	1	-1	-1	1
1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	1
1	-1	1	1	-1	-1	1
-1	1	1	-1	-1	1	1
1	1	-1	-1	1	1	-1
-1	1	-1	1	1	-1	-1

The $\boldsymbol{Q_B}$ criterion

Measures the efficiency to estimate many potential models.

1. Maximal model including the intercept, all main effects and all two-factor interactions.
2. Sub-models of interest satisfy functional marginality.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 + \varepsilon$$

Tsai, Gilmour, & Mead (2007)

The $\boldsymbol{Q_B}$ criterion

Measures the efficiency to estimate many potential models.

1. Maximal model including the intercept, all main effects and all two-factor interactions.
2. Sub-models of interest satisfy functional marginality.
3. A_s criterion to measure the estimation efficiency of sub-models.

$$\boxed{A_s = \sum_{i=1}^{p} \mathrm{Var}\left(\hat{\beta}_i\right)}$$

Tsai, Gilmour, & Mead (2007)

The $\boldsymbol{Q_B}$ criterion

Measures the efficiency to estimate many potential models.

4. Prior probabilities:
   - π_1: Active main effect.
   - π_2: Active interaction given that both of the main effects of the factors involved are active too.
   - π_3: Active interaction given that one of the main effects of the factors involved is active.

   Under this framework, we can calculate the prior probability that sub-model is the best.

Li et al. (2006): $\pi_1 = 0.41$, $\pi_2 = 0.33$ and $\pi_3 = 0.045$

Tsai, Gilmour, & Mead (2007)

The $\boldsymbol{Q_B}$ criterion

Weighted average of the A_s criterion over all sub-models of interest.

Weights: prior probability of a sub-model being the best.

Example I (cont.): 24-run 7-factor designs.

Consider $\pi_1 = 0.41$, $\pi_2 = 0.33$ and $\pi_3 = 0.045$.

Design 1

$$Q_B = 0.246$$

Design 2

$$Q_B = 0.255$$

Minimizing Q_B is equivalent to maximizing the estimation efficiency for the sub-models of the maximal model.

An alternative expression

Let D be an n-run m-factor two-level design and $T = DD^T$ with elements T_{ij} . We define $M_k = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij}^{k}$ (Butler, 2003).

Theorem: For a maximal model including the intercept, all main effects and all two-factor interactions, minimizing the Q_B criterion is equivalent to minimizing

$$w_1 M_1 + w_2 M_2 + w_3 M_3 + w_4 M_4, \qquad (1)$$

where w_k depends on π_1, π_2 and π_3 only.

Computing the Q_B criterion using (1) is computationally cheap!

Outline

1. Introduction
2. Q_B criterion for evaluating experimental designs
3. A metaheuristic algorithm for constructing efficient designs
4. Results and Conclusions

Metaheuristics

A metaheuristic is a high-level problem independent algorithmic framework that provides a set of guidelines or strategies to develop effective optimization algorithms (Sörensen & Glover, 2013).

Local Search

Make small modifications to an existing single solution.

Variable Neighborhood Search
Iterated Local Search

Constructive

Build a single solution from scratch.

Greedy Randomize Adaptive Search
Pilot method

Population-Based

Work with a set of solutions which can communicate or be combined.

Genetic Algorithms
Particle Swarm Optimization

Local Search

- Local search algorithms perform local changes to an existing solution to attempt to find a better one.

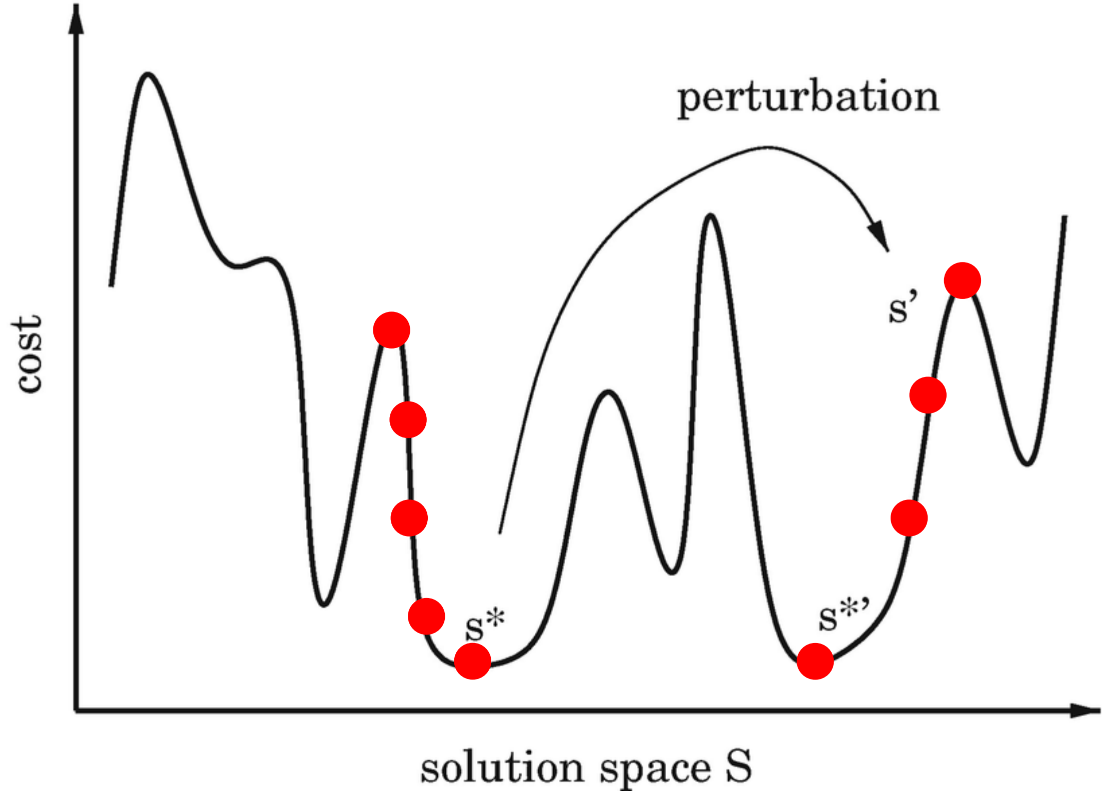
- These algorithms, however, may get stuck in a locally optimal solution instead of the global optimum, since they do not examine all possible changes to the existing solution.

A local search metaheuristic uses an effective strategy to “escape” from this local optimum.

Iterated Local Search (ILS)

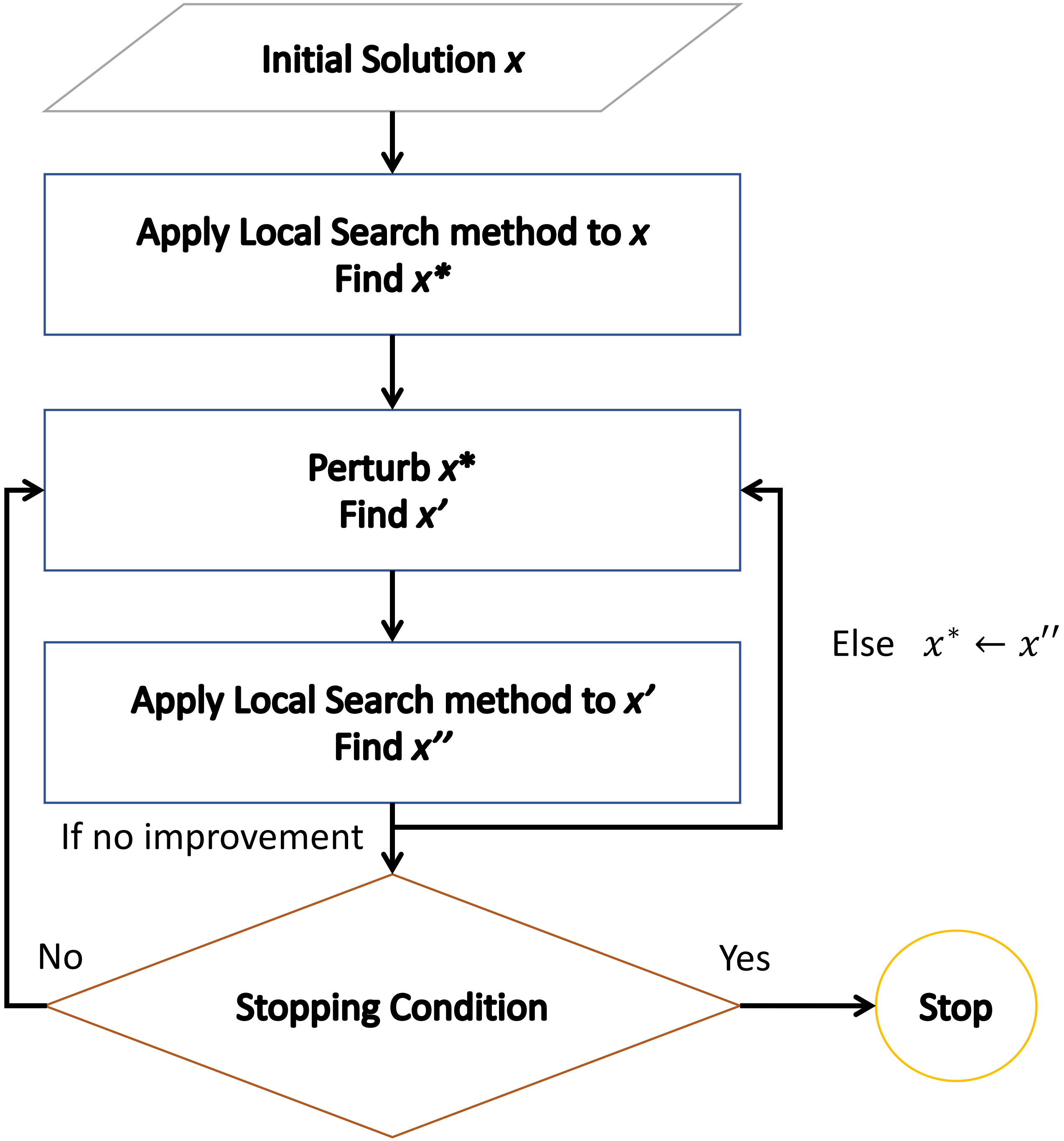
Principles:

- The solution obtained from a local search method is already quite good.
- Perform a small perturbation (modification) to that solution and start from there.



Luorenço et al. (2019)

Basic ILS


Initial Solution x
Apply Local Search method to x
Find x*
Perturb x*
Find x'
Apply Local Search method to x'
Find x''
Else $x^* \leftarrow x''$
If no improvement
No
Stopping Condition
Yes
Stop

Perturbation-Based Coordinate Exchange Algorithm

Constructs two-level designs that minimize the Q_B criterion.

Building Blocks:

1. Local Search: Coordinate-exchange algorithm (Meyer & Nachtsheim, 1995).

2. Perturbation operator.

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.071$

12 runs and 6 factors

$D =$

1	-1	1	1	1	-1
1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.064$

$D =$

-1	-1	1	1	1	-1
1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.053$

$D =$

-1	-1	1	1	1	-1
-1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.051$

$D =$

-1	-1	1	1	1	-1
-1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
-1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.051$

$D =$

-1	-1	1	1	1	-1
-1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
-1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.051$

$D =$

-1	-1	1	1	1	-1
-1	-1	-1	1	-1	1
1	-1	1	1	-1	1
1	-1	-1	1	1	-1
1	1	1	-1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	-1	1
-1	-1	-1	-1	1	1
-1	1	-1	-1	-1	1
1	1	1	-1	1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.023$

$D =$

-1	-1	-1	1	-1	-1
-1	-1	1	1	1	-1
1	-1	1	1	-1	1
1	1	-1	1	1	-1
1	-1	1	1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	1	1
-1	-1	-1	-1	1	1
-1	-1	-1	-1	-1	**-1**
1	-1	1	-1	-1	-1
1	1	-1	-1	-1	1

Coordinate-Exchange Algorithm

- Local Search.
- One move: Sign switch a coordinate in the design *D*.

Example:

$\pi_1 = 0.41, \pi_2 = 0.11, \pi_3 = 0.0$

$Q_B = 0.023$

$$D^* = \begin{matrix} -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 \end{matrix}$$

Perturbation Operator

- Select coordinates at random and sign switch their elements.

- Tuning parameter α denoted the percentage of coordinates that are affected.

$D^* =$

-1	-1	-1	1	-1	-1
-1	-1	1	1	1	-1
1	-1	1	1	-1	1
1	1	-1	1	1	-1
1	-1	1	1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
1	1	-1	1	1	1
-1	-1	-1	-1	1	1
-1	-1	-1	-1	-1	-1
1	-1	1	-1	-1	-1
1	1	-1	-1	-1	1

Perturbation Operator

- Select coordinates at random and sign switch their elements.

- Tuning parameter α denoted the percentage of coordinates that are affected.

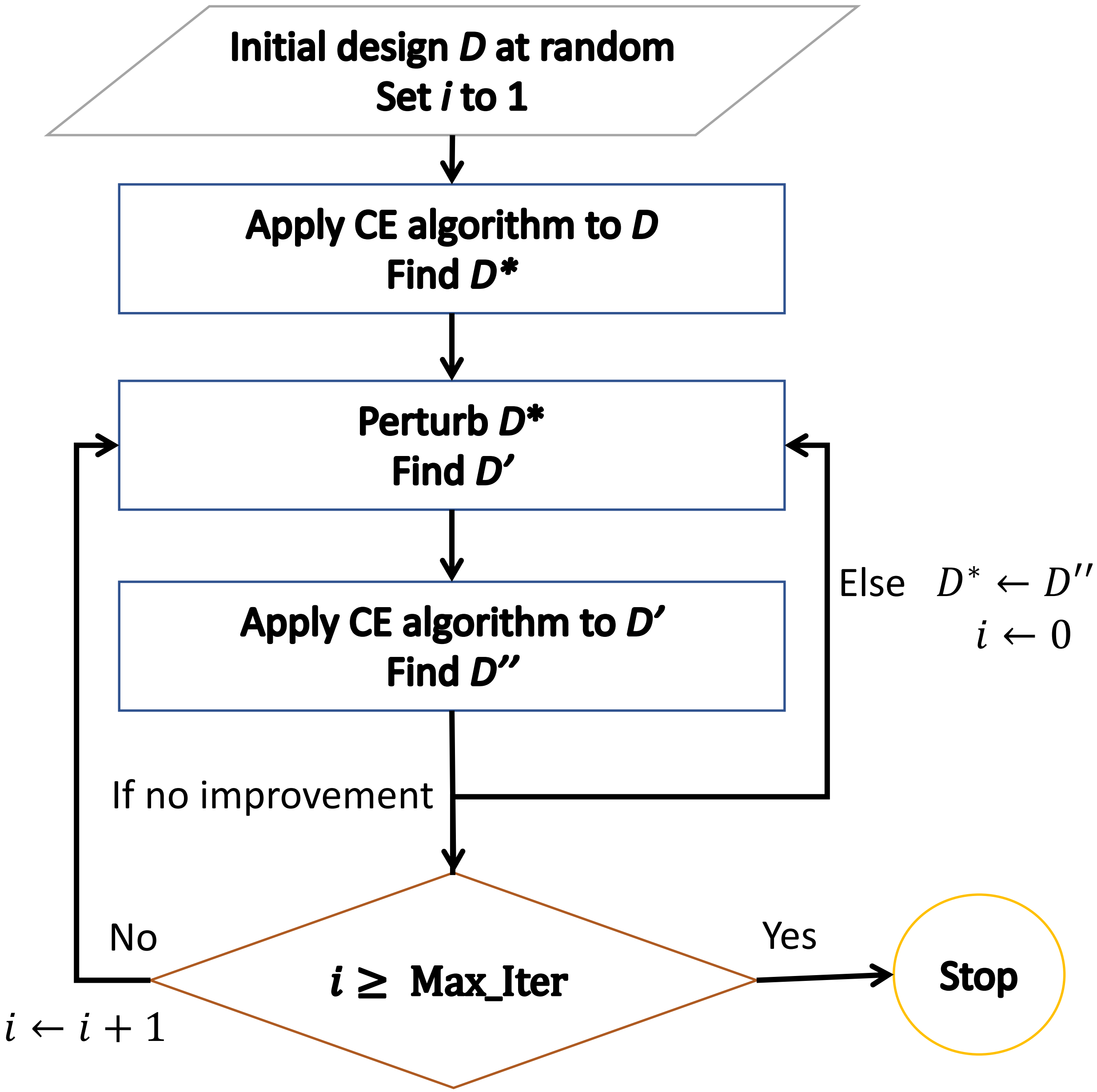
Example:

$$\alpha = 0.05$$

$D' =$

-1	-1	-1	1	-1	-1
-1	-1	1	1	1	-1
1	-1	1	1	-1	1
1	1	-1	1	1	-1
1	-1	1	1	1	1
-1	1	1	-1	1	-1
-1	1	1	-1	-1	1
-1	1	-1	**-1**	1	1
-1	-1	-1	-1	1	1
-1	**1**	-1	-1	-1	**1**
1	-1	1	-1	-1	-1
1	1	-1	-1	-1	1

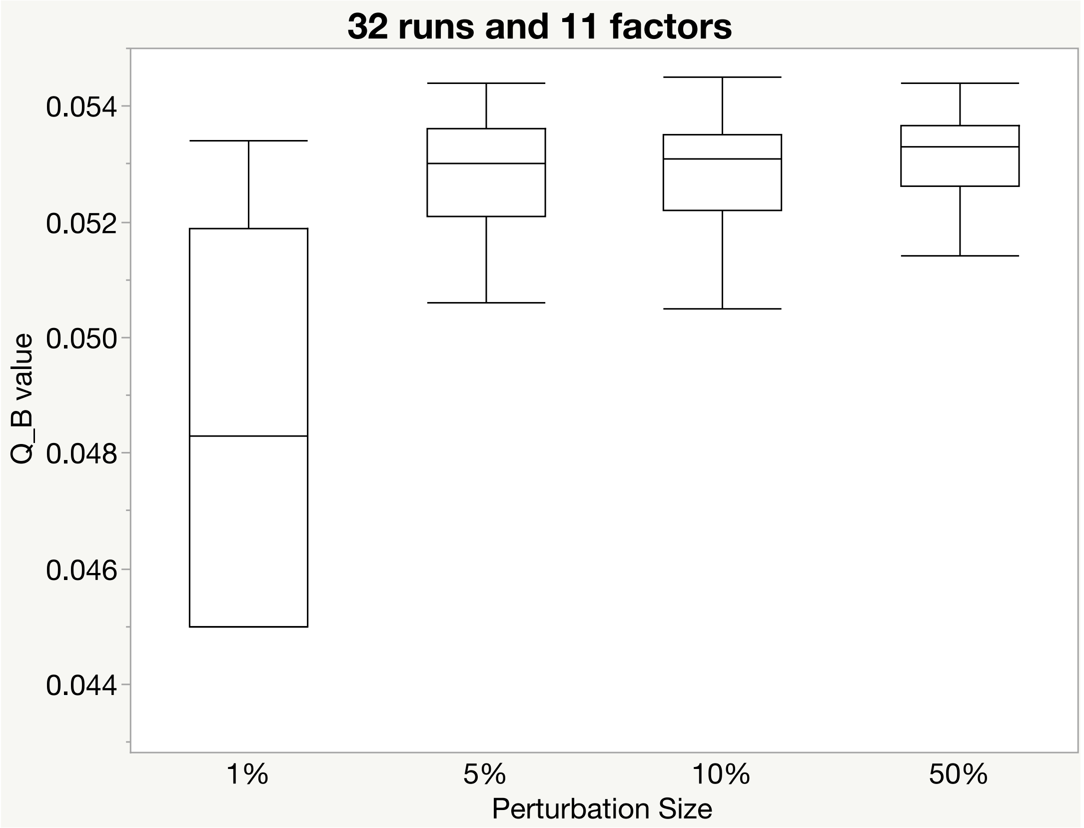
PBCE algorithm in full

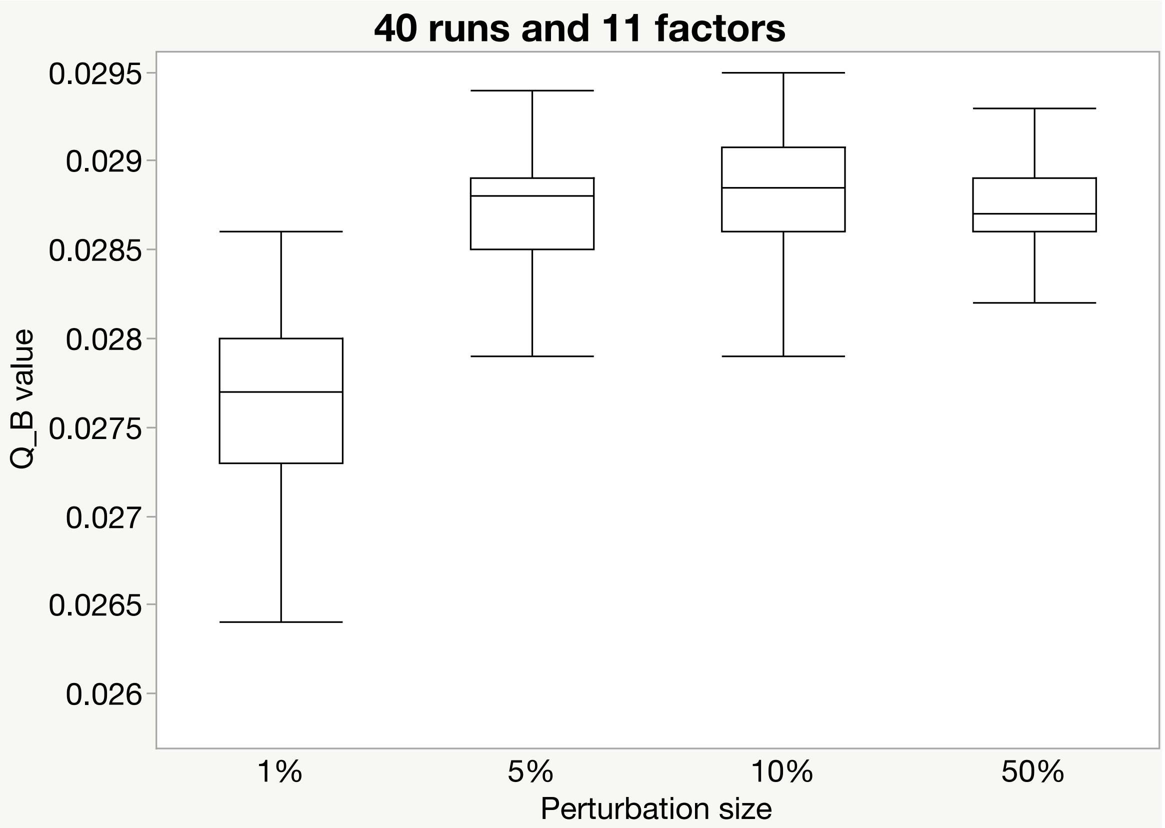


Parameters:
Max_Iter: maximum number of iterations without improvement.
α: perturbation size
Repetitions: number of repetitions of the whole algorithm.

Evaluation Perturbation Size (α)

100 Q_B values obtained using the PBCE algorithm with Max_Iter = 100.





$\pi_1 = 0.5, \pi_2 = 0.4, \pi_3 = 0.0$

Computing Times

10 repetitions of the PBCE algorithm with $\alpha = 0.01$ and Max_Iter = 100.

Factors	Runs	Mean $\pm$ s.d. (seconds)
11	32	10.1 ± 2.7
11	40	18.8 ± 6.3
11	48	27.4 ± 11.0
14	40	11.18 ± 2.7
14	56	65.5 ± 22.4
14	64	101.7 ± 58.1

Implemented in Python v3.7.
Standard CPU with an Intel(R) Core(TM) i7 processor with 2.6Ghz and 16 GB of RAM.

Benchmark Instances

We obtained benchmark instances involving 7 and 11 factors from Mee et al. (2017). We develop larger instances involving 16 factors.

Alternative algorithms:

- Coordinate-exchange algorithm with 1000 iterations (Meyer & Nachtsheim, 1995).
- Restricted columnwise-pairwise algorithm with 1000 iterations (Li, W. 2006).
- Mixed Integer Quadratic Programming with Gurobi and a maximum search time of 20 min.

Our PBCE algorithm involves an $\alpha = 0.01$, Max_Iter = 100, and 5 repetitions.

Results I

Factors	Runs	PBCE Algorithm	Coordinate-Exchange Algorithm	Restricted Columnwise-Pairwise Algorithm	Mixed Integer Quadratic Programming
		$\pi_1 = 0.5, \pi_2 = 0.8$ and $\pi_3 = 0.0$			
7	16	**0.105**	**0.105**	**0.105**	**0.105**
	20	**0.0652**	**0.0652**	**0.0652**	**0.0652**
	24	**0.0333**	**0.0333**	0.0351	**0.0333**
	28	**0.0203**	**0.0203**	**0.0203**	**0.0203**
	32	**0.0075**	**0.0075**	**0.0075**	**0.0075**
		$\pi_1 = 0.5, \pi_2 = 0.4$ and $\pi_3 = 0.0$			
11	20	**0.1726**	**0.1726**	0.1879	0.2154
	24	**0.0917**	**0.0917**	0.1189	0.1408
	32	**0.0478**	0.0531	0.0560	0.0664
	40	**0.0268**	0.0286	0.0318	0.0371
	48	**0.0161**	0.0168	0.0187	0.0235

Smaller the better

Results II

Factors	Runs	PBCE Algorithm	Coordinate-Exchange Algorithm	Restricted Columnwise-Pairwise Algorithm	Mixed Integer Quadratic Programming
$\pi_1 = 0.41, \pi_2 = 0.33$ and $\pi_3 = 0.045$					
16	32	**0.0808**	**0.0808**	0.1374	-
	40	**0.0548**	0.0607	0.0794	-
	48	**0.0369**	0.0417	0.0519	-
	56	**0.0266**	0.0323	0.0356	-
	64	**0.0213**	0.0230	0.0256	-

Smaller the better

Discussion of PBCE algorithm

- PBCE algorithm is the first dedicated algorithm to construct large designs that minimize the Q_B criterion.

- For large numbers of factors, our PBCE algorithm outperforms alternative algorithmic strategies.

- For up to 7 factors, our mixed integer quadratic programming approach is attractive since it can provide certificates of optimality.

Vazquez, A. R., Wong, W.-K., Goos, P. (2021). Constructing two-level Q_B-optimal designs for screening experiments using mixed integer programming and heuristic algorithms. In preparation.

Outline

1. Introduction
2. Q_B criterion for evaluating experimental designs
3. A metaheuristic algorithm for constructing efficient designs
4. Results and Conclusions

Alternative designs for the tuberculosis inhibition experiment

- We constructed alternative 14-factor designs with 64, 80, 96 and 112 runs.

- Using the PBCE algorithm, we generated designs which optimize the Q_B criterion with $\pi_1 = 0.41$, $\pi_2 = 0.33$ and $\pi_3 = 0.045$ (Li et al., 2006).

- We evaluate the performance of these designs using a simulation study.

Available designs in the literature

- 14-factor Bayesian D-optimal designs with 64, 80, 96 and 112 runs (DuMouchel & Jones, 1994).

- 14-factor D-optimal design with 112 runs (Atkinson et al., 2007).

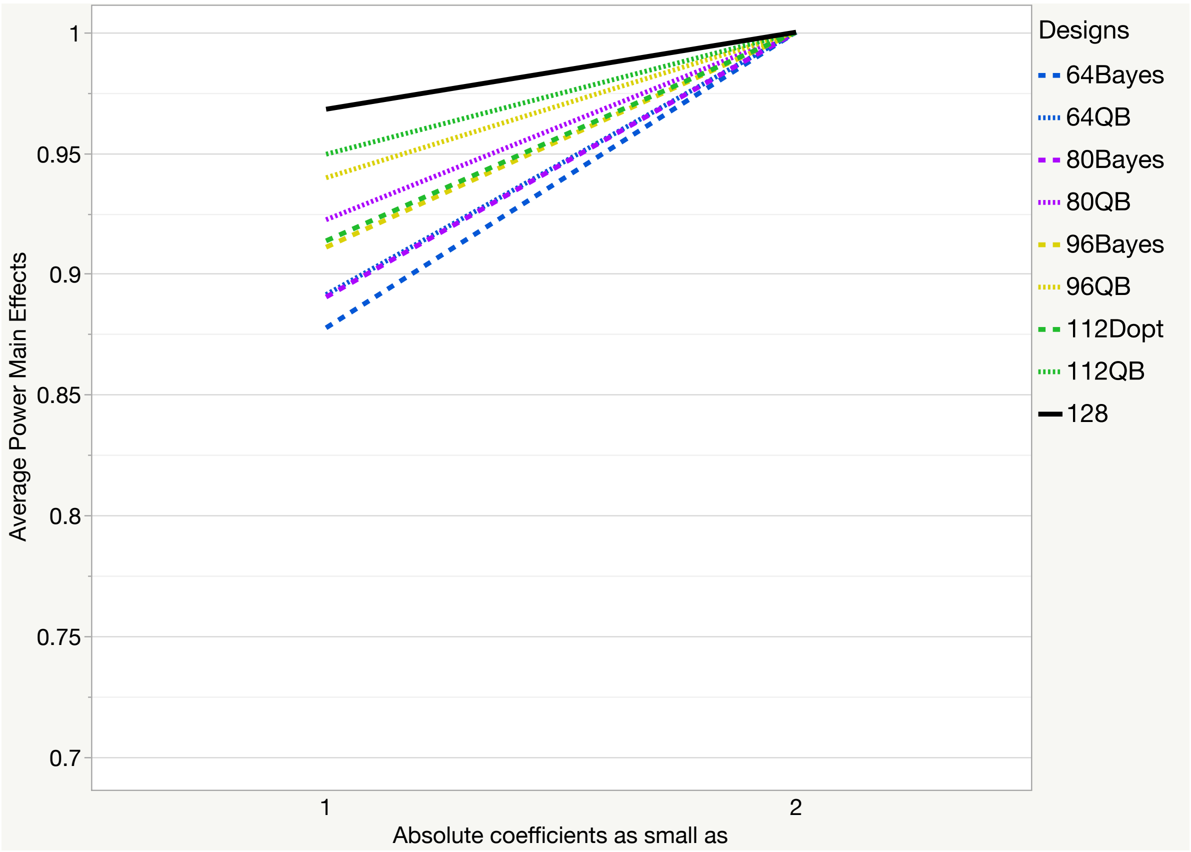
- 14-run design with 128 runs (Silva et al. , 2016).

Simulation Study

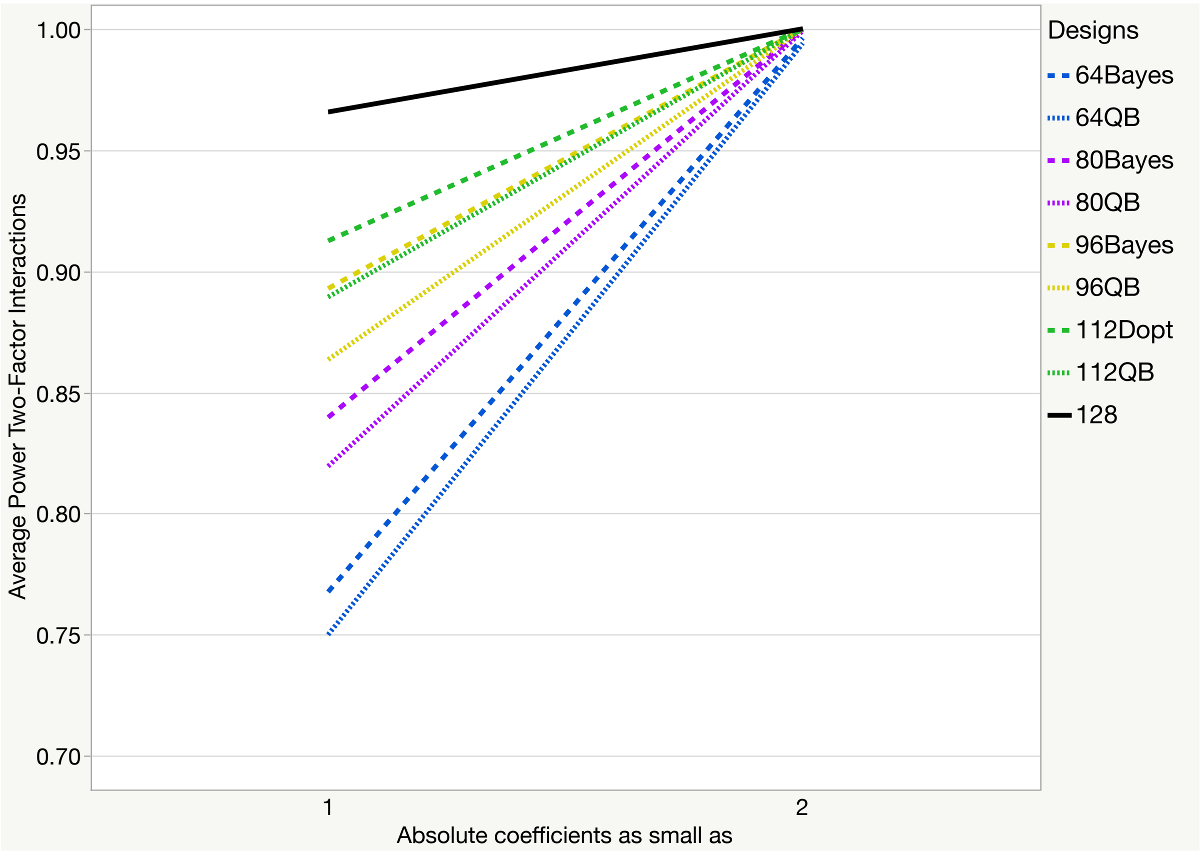
1. We randomly selected 8 main effects and 6 two-factor interactions (satisfying weak effect heredity) as active.
2. We obtained coefficients for the active effects by adding a 1 or a 2 to an exponentially distributed random number. A ‘$+$’ or ‘$-$’ was randomly assigned.
3. The coefficients for the 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

Main Effects



Two-Factor Interactions



Power: Proportion of active effects that are successfully detected.

Conclusions

- The tuberculosis inhibition experiment would have benefited from a smaller design.
- Metaheuristics allow the development of effective algorithms to construct attractive experimental designs.
- Two-level designs can be used to find the right hyperparameter values of machine learning algorithms (Choueki et al., 1997; Lujan-Moreno, 2018).
- Future research: Explore other metaheuristics such as particle swarm optimization to construct Q_B-optimal designs with two or more levels.

References

- Silva, A., Lee, B.-Y., Clemens, D. L., Kee, T., Ding, X., Ho, C.-M., and Horwitz, M. A. (2016). Output-driven feedback system control platform optimizes combinatorial ther- apy of tuberculosis using a macrophage cell culture model. *Proceedings of the National Academy of Sciences*, 113:E2172–E2179.
- Miller, A., and Sitter, R. R. (2001). Using the folded-over Plackett-Burman designs to consider interactions. Technometrics, 43, 44-55.
- Tsai, P.-W., Gilmour, S. G., and Mead, R. (2007). Three-level main-effects designs ex- ploiting prior information about model uncertainty. *Journal of Statistical Planning and Inference*, 137:619–627.
- Butler, N. A. (2003). Minimum aberration construction results for nonregular two-level fractional factorial designs. *Biometrika*, 90:891–898.
- Sörensen, K. and Glover, F. (2013). *Metaheuristics*. Encyclopedia of Operations Research and Management Science, pages 960-970. Springer.
- Li, X., Sudarsanam, N., and Frey, D. D. (2006). Regularities in data from factorial experiments. *Complexity*, 11:32–45.
- Lourenço, H. R., Martin, O. C., and Stützle, T. (2019). *Iterated local search: Framework and applications*. Handbook of Metaheuristics, pages 129–168. Springer.
- Meyer, R. K. and Nachtsheim, C. J. (1995). The coordinate-exchange algorithm for con- structing exact optimal experimental designs. *Technometrics*, 37:60–69.
- Li, W. (2006). Screening designs for model selection. In A., D. and S., L., editors, *Screening*, pages 207–234. Springer, New York, NY.
- DuMouchel, W. and Jones, B. (1994). A simple Bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36:37–47.
- Choueki, M. J., Mount-Campbell, C. A., and Ahalt, S. C. (1997). Building a 'quasi optimal' neural network to solve the short-term load forecasting problem. *IEEE Transactions on Power Systems*, 12:1432-1439.
- Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G., and Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters with a random forest case-study. *Expert Systems with Applications,* 109:195-205