Comparison of Two Strategies of Screening Experiments: Single-shot Experiment vs. Two-stage Screening Experiment

Kelvin Njuki

July 2021

Abstract

Experiments involving many factors are often complex, time-consuming, and expensive. Screening out the least important factors helps the experimenter(s) allocate the limited resources efficiently to the most important factors. Supersaturated and orthogonal array designs are among the designs used to conduct screening experiments. Supersaturated designs ($SSDs$) are those where the number of runs (observations) is less than the number of factors, while orthogonal array (OA) designs are those where at least the columns are orthogonal to each other. In this study, we conduct a simulation study to compare two strategies of screening experiments. Strategy one is a single-shot experiment using an orthogonal array. Strategy two is a two-stage screening experiment that involves supersaturated design in the first stage and a follow-up using orthogonal array design in the second stage. The study investigates two models: (1) a model with a subset of main effects being active and (2) a model with a subset of main effects and some two-factor interaction effects active. The two strategies are analyzed via the Dantzig selector method. The power to detect active effects, type I error rate, and false discovery rate are computed and compared. Generally, strategy one performs better than strategy two. When effect sparsity is high, the two strategies are comparable.

Keywords: Supersaturated designs (SSDs), orthogonal array (OA) designs, Dantzig selector, Strategy, Power, Type I Error, False Discovery Rate (FDR).

---

1 Introduction

The interest for studying large systems by the experimenters is increasing as the world advances to a higher level in science and technology. The resources to perform large experiments (experiments involving many factors) may be inadequate in most cases. Further, these experiments are complex and time-consuming. In such a scenario, screening out the least important factors would help the experimenter(s) allocate the available resources efficiently to the most important factors. Various strategies/designs can be used to conduct screening experiments. However, an experimenter would prefer a strategy that has high power for detecting the active factors, low type I error rate and a low false discovery rate. Throughout the paper, we use the term strategy to refer to an approach for conducting a screening experiment. This study is comparing two strategies;

Strategy 1: A single-shot experiment using orthogonal array designs: OA experiment.

Strategy 2: A two-stage screening experiment using supersaturated designs (SSDs) in the first stage and orthogonal array designs in the second stage (follow-up): SSD + OA.

We start by defining some notations and models that are important for a better understanding of this paper. Consider $\mathbf{Y}$ as an n x 1 vector of the response, $\mathbf{X}$ as an n x $(p+1)$ model matrix ($p = k$ or $\binom{k}{2}$), $\boldsymbol{\beta} = (\beta_0, ..., \beta_k)^T$ as a vector of unknown parameters, and $\boldsymbol{\epsilon} \sim N(0,1)$ is a vector of independent errors. n and k are the number of runs and factors respectively. Assume each factor has two levels (-1 and $+1$). In this study, we considered two main models.

$$\mathbf{Y} = \mathbf{X}_a \boldsymbol{\beta}_a + \boldsymbol{\epsilon} \tag{1}$$

$$\mathbf{Y} = \mathbf{X}_a \boldsymbol{\beta}_a + \mathbf{X}_{a.i} \boldsymbol{\beta}_{a.i} + \boldsymbol{\epsilon} \tag{2}$$

where $\mathbf{X}_a$ is an n x a model matrix for subset of main effects active (a), $\mathbf{X}_{a.i}$ is an n x $(a.i)$ model matrix for some two-factor interaction effects active $(a.i)$, $\boldsymbol{\beta}_a$ is an ax1 vector of coefficients for the subset of main effects active, $\boldsymbol{\beta}_{a.i}$ is an $(a.i)$ x 1 vector of coefficients for some two-factor interaction effects active and $\boldsymbol{\epsilon} \sim N(0,1)$ is an n x 1 vector of independent errors.

We define a supersaturated design $\mathbf{D}$ as a factorial design, with the number of observations/runs (n) less than the number of factors (k). These designs were introduced by Satterthwaite (1959), who suggested constructing them randomly. They are used where experimenters are willing to adhere to the effect sparsity assumption (only a few factors are important in the experiment, Box and Meyer (1986)). There have been improvements in these designs by researchers since then.

Booth and Cox (1962) improved these random designs by ensuring all design columns were nearly orthogonal to each other. Their criterion selects designs that minimizes

$$E(s^2) = \frac{2}{k(k-1)} \sum_{2 \leq i < j} s_{ij}^2, \tag{3}$$

where s_{ij} is the $(i,j)th$ element of $\mathbf{X'X}$. Few years later, Lin (1993) introduced a new class of supersaturated designs constructed via half fractions of Hadamard matrices. In the same year, Wu (1993) proposed constructing $E(s^2)-$optimal designs by augmenting Hadamard matrices with two-factor interaction columns. Li and Wu (1997) built $E(s^2)-$optimal SSDs based on a $D-$optimal design search by applying columnwise-pairwise algorithms. Marley and Woods (2010) extended

$E(s^2)$ criterion by Booth and Cox (1962) to include the intercept column of $\mathbf{X}$, such that,

$$E(s^2) = \frac{2}{k(k-1)} \sum_{1 \le i < j} s_{ij}^2. \tag{4}$$

Equations **3** and **4** are basically the same when the design is balanced. Weese, Smucker, and Edwards (2015) introduced Unbalanced $E(s^2)-$optimal designs which relaxes the balanced condition when constructing $E(s^2)-$optimal designs. Further, they examined $Var(s)-$optimal designs. They noted that a design chosen to minimize $Var(s) = E(s^2) - E(s)^2$ would allow very high s values with little or no variation among them. Therefore, they explored Constrained $Var(s)-$optimal designs which minimizes $Var(s)$ subject to a specified $E(s^2)$ efficiency. The efficiency for design $\mathbf{D}$ is defined as

$$E(\mathbf{D}) = \frac{E(s^2)(\mathbf{D}^*)}{E(s^2)(\mathbf{D})}, \tag{5}$$

where $\mathbf{D}^*$ is the $E(s^2)-$optimal designs (balanced or unbalanced).

Weese, Edwards, and Smucker (2017) revisited and refined the Constrained $Var(s)-$optimal designs by imposing an additional constraint [the Constrained $Var(s)-$optimal designs must have a positive average columns correlation]. They referred the new designs as **Constrained Positive Var(s)−optimal designs**. They discovered that when the signs of the main effects in these designs are specified in advance, the designs have a higher power of detecting the active factors. Weese, Edwards, and Smucker (2017) found that when these designs (Constrained Positive $Var(s)-$optimal designs) are paired with the Dantzig Selector (Candes and Tao (2007)), they reasonably control type I error and increase power. Therefore, in this study, we used the Constrained Positive $Var(s)-$optimal designs and analyzed them via Dantzig Selector. Below we outline the steps for constructing the Constrained Positive $Var(s)-$optimal designs as outlined by Weese, Edwards, and Smucker (2017), which uses coordinate exchange algorithm by Meyer and Nachtsheim (1995).

(i) Construct n x k initial designs randomly until you find one which has $E(s)$ being greater than zero.

(ii) Construct an initial design that satisfies

$$E_{E(s^2)} = \frac{E(s^2)(D*)}{E(s^2)(D)} > c, E(s) > 0 \tag{6}$$

where $E(s^2)$ is as defined in **3**, D^* is the $E(s^2)-$optimal design and c is a user-specified efficiency that determines how close to $E(s^2)-$optimal the design must be. Weese, Edwards, and Smucker (2017) found that $c = 0.8$ provides a balance between type I error and power. Iterate from one coordinate to the next, row by row, and if the last row is reached, begin at the first row again. At each coordinate, consider the impact on the design when the current coordinate value is exchanged. Make an exchange when $E_{E(s^2)}$ is increased while $E(s^2) > 0$ is maintained. When $E_{E(s^2)} \ge c$, go to Step *(iii)*.

(iii) Repeat this from one coordinate to the other and one row to the other. When you get to the last row, start again at the first row. Every time, evaluate the changes in the $E(s^2)-$efficiency, $Var(s)$ criterion, and $E(s)$ when you multiply the current coordinate value by -1.

If there will be an improvement of $Var(s)$ while maintaining $E(s) > 0$ and $E_{E(s^2)} \ge c$ after the coordinate exchange, adopt the exchange, else proceed to the immediate next coordinate.

Keep working on this step, running it repeatedly through the design until a local optimum convergence is achieved. It is recommendable to run several algorithm tries from different initial designs for efficient designs.

The second type of design in this study is an orthogonal array. An orthogonal array is a type of design whose columns are at least orthogonal to each other. Formally, Hedayat, Sloane, and Stufken (1999) defines orthogonal array as a N x k array A with entries from S with s levels, strength t and index λ (for some t in the range $0 \leq t \leq k$), if every N x t sub-array of A contains each $t-$tuple based on S exactly λ times as a row. They are denoted by $OA(N, k, s, t)$ where N, k, s, t and λ are the parameters of the orthogonal array. The parameters are defined as follows, N is the number of runs in the design, k represent the number of factors, s is the number of factor levels, S is a set of s levels, t is the strength of the design, and λ is the number of times a pair exists in N x t sub-array. Note, this notation is not universal.

Orthogonal array designs were first introduced in the 1940s by Rao (1947) as a combinatorial arrangement with applications to statistics. Since then, these designs have been playing a significant role in multi-factor experiments. Dey and Mukerjee (2009) discussed the construction and optimality of orthogonal arrays as fractional factorial designs. Extensive literature about orthogonal array designs can be found in Hedayat, Sloane, and Stufken (1999).

This study considered strength two orthogonal array designs where columns have two levels (-1 and $+1$). The strength of an orthogonal array design is directly related to its resolution. The resolution is always one higher than the strength. Thus, the strength 2 OA designs in this study are of resolution III. Grömping (2018) developed **DoE.base** `R` package, which was an excellent resource for this study. We generated the OA designs from this package using the `oa.design` function. The package relies heavily on a catalog of orthogonal arrays, with most of them coming from Kuhfeld (2010).

Supersaturated designs presently are viewed with much suspicion to be widely used. In a small survey conducted by Weese et al. (2020), only 9.5% indicated that they use SSDs regularly. Most respondents reported using fractional factorial designs (e.g., orthogonal array), among others. The researchers noted that some practitioners considered SSDs as a one-shot experiment, which did not guarantee to find the few important factors. Therefore, evaluating a sequential experimentation strategy starting with a supersaturated design could increase the practitioners' confidence in identifying the active factors. However, there is scant literature about following up a supersaturated design. Gutman et al. (2014) suggested a follow-up approach based on Bayesian $D-$optimality. A collaborative research *(Powerful Regularization-Based Screening Experimentation for Process Optimization with Applications to Additive Manufacturing)* suggested foldover and semi-foldover augmentation of the SSDs as a follow-up approach. However, there is scant literature on augmentation. This study investigates a follow-up approach that uses orthogonal array design with columns equivalent to the active factors identified in the first analysis stage.

2 Analysis and Simulation

2.1 Analysis Method

Weese, Edwards, and Smucker (2017) found that when Constrained Positive $Var(s)-$optimal designs are paired with the Dantzig Selector, they reasonably control type I error and increase power. Further, Marley and Woods (2010) demonstrated the effectiveness of this analysis method. Therefore, in this study, we used the Dantzig selector analysis method proposed by Candes and Tao (2007). The method is commonly used to estimate parameters when the number of observations is not more than the number of factors. The estimator $\hat{\boldsymbol{\beta}}$ is a solution to the ℓ_1- regularization problem

$$argmin_{\hat{\beta}\in\mathbb{R}^k}||\hat{\boldsymbol{\beta}}||_{\ell_1} \tag{7}$$

subject to

$$||(\mathbf{X}'\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})||_{\ell_\infty} \leq \delta$$

where $\mathbf{y}$ is an n x 1 vector of observations, $\mathbf{X}$ is an n x k model matrix, $||\boldsymbol{\beta}||_{\ell_1} = |\beta_0|+|\beta_1|+...+|\beta_m|$ is the ℓ_1-norm and $\delta \geq 0$ is a tuning parameter. The estimator requires one to specify the tuning parameter (δ). Here, we selected the tuning parameter following Marley and Woods (2010), via the BIC statistic given by

$$BIC = n\log(\frac{(\mathbf{y}-\mathbf{X}\hat{\beta})'(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})}{n}) + p\log(n) \tag{8}$$

where p is the number of model terms and $\hat{\boldsymbol{\beta}}$ is a vector of the least-squares estimates found by regressing the response on the set of factors deemed active by the Dantzig selector.

The version of the Dantzig selector used in our study was discussed by Phoa, Pan, and Xu (2009) as a standard linear program. That is;

min $c'\chi$ subject to $A\chi \geq b$ and $\chi \geq 0$ where

$$c = \begin{pmatrix} \mathbf{1}_k \\ \mathbf{0}_k \end{pmatrix}, A = \begin{pmatrix} \mathbf{X}'X & -\mathbf{X}'\mathbf{X} \\ -\mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{X} \\ 2\mathbf{I}_k & -\mathbf{I}_k \end{pmatrix}, b = \begin{pmatrix} -\mathbf{X}'\mathbf{y}-\delta\mathbf{1}_k \\ \mathbf{X}'\mathbf{y}-\delta\mathbf{1}_k \\ \mathbf{0}_k \end{pmatrix}, \chi = \begin{pmatrix} u \\ u+\beta \end{pmatrix} \tag{9}$$

.

In our study, we solved the above linear program to obtain the Dantzig selector $\hat{\boldsymbol{\beta}}$ for some values of δ ranging from 0 to δ_0. To obtain δ_0 we centered the response variable $\mathbf{Y}$ and centered and scaled the columns of $\mathbf{X}$ to have zero mean and unit variance. We computed $\delta_0 =$ max $|x_i'y|$ where x_i is the ith column of $\mathbf{X}$.

To classify an effect as active, we needed a threshold γ such that if $|\hat{\beta}_j| > \gamma$, the corresponding effect is classified as active. An ideal approach for threshold in strategy 1 and strategy 2 stage 2 would be $\gamma = \sigma$ (Weese, Edwards, and Smucker (2017)), while in strategy 1 stage 1, a data-driven approach suggested by Phoa, Pan, and Xu (2009) ($\gamma = 0.1$*Max$|\hat{\beta}_j|$) would be appropriate. In our study, we conducted a grid search to find the best thresholds (See a snapshot of the grid search results below). Complete results are in the appendix.

Direction	SN	Sparsity Level	Designs Infomation	$0.5 * \hat{\sigma}$	$\hat{\sigma}$	$1.5 * \hat{\sigma}$	$0.1 * Max\|\hat{\beta}_j\|$	$0.05 * Max\|\hat{\beta}_j\|$	$0.25 * Max\|\hat{\beta}_j\|$	$0.5 * Max\|\hat{\beta}_j\|$
0.0	0.0	0.25	N=28,n1=12,k=16	0.4907749	0.9815497	1.472325	0.3533552	0.1766776	0.8833881	1.766776
0.5	0.0	0.25	N=28,n1=12,k=16	0.4936114	0.9872228	1.480834	0.4054413	0.2027207	1.0136033	2.027207
0.0	0.5	0.25	N=28,n1=12,k=16	0.4964188	0.9928375	1.489256	0.4515195	0.2257598	1.1287988	2.257598
0.5	0.5	0.25	N=28,n1=12,k=16	0.4958810	0.9917620	1.487643	0.4434192	0.2217096	1.1085481	2.217096
0.0	0.0	0.50	N=28,n1=12,k=16	0.4971232	0.9942464	1.491370	0.5125640	0.2562820	1.2814100	2.562820
0.5	0.0	0.50	N=28,n1=12,k=16	0.4873365	0.9746731	1.462010	0.3542021	0.1771011	0.8855053	1.771011
0.0	0.5	0.50	N=28,n1=12,k=16	0.4951605	0.9903210	1.485481	0.4007086	0.2003543	1.0017716	2.003543
0.5	0.5	0.50	N=28,n1=12,k=16	0.4959387	0.9918774	1.487816	0.4521361	0.2260681	1.1303403	2.260681
0.0	0.0	0.75	N=28,n1=12,k=16	0.4972456	0.9944912	1.491737	0.4380976	0.2190488	1.0952441	2.190488
0.5	0.0	0.75	N=28,n1=12,k=16	0.4979111	0.9958222	1.493733	0.5201416	0.2600708	1.3003540	2.600708
0.0	0.5	0.75	N=28,n1=12,k=16	0.4877675	0.9755349	1.463302	0.4061687	0.2030843	1.0154216	2.030843
0.5	0.5	0.75	N=28,n1=12,k=16	0.4929019	0.9858037	1.478706	0.4503304	0.2251652	1.1258260	2.251652
0.0	0.0	0.25	N=48,n1=16,k=24	0.4942238	0.9884475	1.482671	0.5056138	0.2528069	1.2640346	2.528069
0.5	0.0	0.25	N=48,n1=16,k=24	0.4953989	0.9907977	1.486197	0.4949026	0.2474513	1.2372565	2.474513
0.0	0.5	0.25	N=48,n1=16,k=24	0.4980659	0.9961318	1.494198	0.5674757	0.2837378	1.4186892	2.837378
0.5	0.5	0.25	N=48,n1=16,k=24	0.4891459	0.9782919	1.467438	0.4002329	0.2001164	1.0005822	2.001164
0.0	0.0	0.50	N=48,n1=16,k=24	0.4954884	0.9909768	1.486465	0.4470984	0.2235492	1.1177460	2.235492
0.5	0.0	0.50	N=48,n1=16,k=24	0.4957008	0.9914016	1.487102	0.4972640	0.2486320	1.2431599	2.486320
0.0	0.5	0.50	N=48,n1=16,k=24	0.4983037	0.9966073	1.494911	0.4884286	0.2442143	1.2210715	2.442143
0.5	0.5	0.50	N=48,n1=16,k=24	0.4982892	0.9965784	1.494868	0.5641203	0.2820602	1.4103009	2.820602

Figure 1: A snapshot of the grid search to find the best thresholds

After the grid search, we settled for $\gamma = 0.5 * \hat{\sigma}$ threshold for strategy 1 and strategy 2 stage 2. In strategy 2 stage 1, we could not estimate σ due to inadequate degrees of freedom. Therefore, we used a data-driven threshold in this stage ($\gamma = 0.05{*}\text{Max}|\hat{\beta}_j|$). The threshold followed a suggestion by Phoa, Pan, and Xu (2009), but with a slight modification.

2.2 Simulation

2.2.1 Simulation Protocol

In this study, we have two main strategies as introduced in Section 1. In each strategy, we have two models; **1** and **2** (defined in Section 1). We considered five pairs of designs in both strategies (Table 1).

Table 1: Designs Properties

OA Runs (N)	SSD Runs (n1)	Factors
28	12	16
48	16	24
52	20	24
68	20	32
76	28	32

In each pair, the number of factors for the two designs was the same while the runs in OA design N (total budget) were more than the runs in the SSDs (n_1). In the follow-up design, we had a scenario where the runs (n_2) were $n_2 = N - n_1$ and a scenario where n_2 would vary with the number of active effects found in stage one. We generated the OA designs from **DoE.base** package in `R` (Grömping (2018)), while the SSDs came from Weese et al. (2020) supplementary materials.

There were two effects directions in each strategy, known and unknown (Weese et al. (2020)) and two signal-to-noise ratio values 0 and 0.5. The signal-to-noise ratio basically shifts the mean of the coefficients. We considered three sparsity levels according to Marley and Woods (2010) when using model **1** as the true model, *i.e.* $0.25n_1$, $0.5n_1$ and $0.75n_1$, where n_1 is the number of runs in the SSDs. In model **2**, two sparsity levels for active main effects (0.3 and 0.4) and two corresponding levels for two-factor interaction effects (0.02 and 0.03) were factored. Findings from Li, Sudarsanam, and Frey (2006) paper guided our selection of these sparsity levels. The paper conducted a meta-analysis on regularities in data from factorial experiments and found that $41\%(36-46)$ of the main effects are active, $11\%(9-14)$ of the two-factor interactions are active. Therefore, for main effects, we chose values that are below and close to 41% (0.3 and 0.4), while for the two-factor interactions effects, we were guided by $Fig.4$ in Li, Sudarsanam, and Frey (2006) and settled on the above values (0.02 and 0.03). The figure suggests that the number of active two-factor interactions is roughly one-third of the active main effects. Further, we considered strong and weak effects' heredity. Basically, we had **12** scenarios for model **1** and **16** for model **2** (Tables 1 and 2).

Table 2: Model 1 scenarios

Direction	SN	Sparsity Level
Known	0.0	0.25
Unknown	0.0	0.25
Known	0.5	0.25
Unknown	0.5	0.25
Known	0.0	0.50
Unknown	0.0	0.50
Known	0.5	0.50
Unknown	0.5	0.50
Known	0.0	0.75
Unknown	0.0	0.75
Known	0.5	0.75
Unknown	0.5	0.75

Table 3: Model 2 scenarios

Direction	SN	Heredity	Active ME	Active 2FIs
Known	0.0	Weak	0.3	0.02
Unknown	0.0	Weak	0.3	0.02
Known	0.5	Weak	0.3	0.02
Unknown	0.5	Weak	0.3	0.02
Known	0.0	Strong	0.3	0.02
Unknown	0.0	Strong	0.3	0.02
Known	0.5	Strong	0.3	0.02
Unknown	0.5	Strong	0.3	0.02
Known	0.0	Weak	0.4	0.03
Unknown	0.0	Weak	0.4	0.03
Known	0.5	Weak	0.4	0.03
Unknown	0.5	Weak	0.4	0.03
Known	0.0	Strong	0.4	0.03
Unknown	0.0	Strong	0.4	0.03
Known	0.5	Strong	0.4	0.03
Unknown	0.5	Strong	0.4	0.03

2.2.2 Generation of the coefficients and the response

We generated the active main effects coefficients from an exponential distribution (Weese et al. (2020)) with a mean of 0.5. We added noise of 0 and 0.5. The coefficients remained positive if the direction of the effects was known; otherwise, the signs were randomly set to -1 and $+1$ with a probability of 0.5. Where inactive effects were considered (results are in the appendix), the coefficients were randomly drawn from $|N \sim (0, \frac{1}{36})|$ (Weese et al. (2020)), otherwise they were assumed to have zero coefficient. The response $\mathbf{Y}$ was generated following the models **1** and **2** (Section 1).

2.2.3 Follow-up Experiment

In strategy 2, the number of factors found active by the Dantzig selector in stage one of the analysis were used to generate a follow-up OA design. Two directions were followed on the number of runs for this follow-up design (n_2): (i) Fixed n_2 - we considered n_2 as the difference between OA design runs and SSD runs ($n_2 = N - n_1$) in each pair, (ii) Varying n_2 - we allowed the function `oa.design` [**DoE.base** package] to choose the appropriate number of runs (n_3) based on the number of active factors found. For clarity, see the structure of our study below.

Strategy 1: Use OA designs.

Strategy 2: (i) Use SSD + Follow-up OA design with ($n_2 = N - n_1$) runs.
(ii) Use SSD + Follow-up OA design with n_3 runs determined by the number of factors found active.

Both directions were implemented in model **1**, while in model **2**, only direction (i) was implemented due to the long run time experienced by this algorithm implementing model **2**. Where direction (ii) was implemented, we tracked the number of runs saved *i.e* ($n_4 = n_2 - n_3$). After design generation, to ensure no change in the model, we used the same coefficients for the active effects as used in stage one (We extracted the corresponding coefficients for these found active effects from the coefficients generated in stage one of the analysis). We generated the response according to model **2** above and used the Dantzig selector to select the active effects. Where n_2 was fixed, a similar threshold to that in strategy one was used ($\gamma = 0.5 * \hat{\sigma}$). But when n_2 was varying, a threshold of 0.5 was used because we could not use $\gamma = 0.5 * \hat{\sigma}$ due to the inability to estimate σ in some designs that had inadequate degrees of freedom. The decision to use 0.5 was arrived at after conducting a grid search (see snapshot in Figure 1), which reported values for $\gamma = 0.5 * \hat{\sigma}$ on a range of $0.47 - 0.52$.

We ran $5,000$ iterations fr model 1 and $1,000$ for model 2. The power, type I error and FDR were computed and visualized.

3 Simulation Results

The results presented in this section are based on the simulation and the analysis method discussed in Section 2. The results show the three metrics (power, type I error, and false discovery rate) against the sparsity levels. Here, we define power as the average proportion of correctly identified active effects (Weese, Smucker, and Edwards (2015)), type I error as the average proportion of inactive effects identified as active (Weese, Smucker, and Edwards (2015)), while false discovery rate is the average proportion of inactive effects classified as active (Weese et al. (2020)). The results are for the two models (Section 1) and scenarios discussed in Section 2.2.1 (Model **2** and **3**).

3.1 Power Results

In Figure 2 (a), we see that strategy 1 screens out the least important effects more effectively than strategy 2. While this seems the case across different sparsity levels, we notice that the two strategies perform relatively the same when sparsity level is high. If the runs in the follow-up designs in strategy 2 are allowed to vary with the active effects found by the Dantzig selector (Figure 2 (b)), we notice similar performance as to when the runs are fixed (Figure 1 (a)). The performance difference between the two strategies is narrow when the directions of the effects are known.

Figures 3, 4 and 5, presents results from model **2**. The results are relatively poor compared to model **1** results. The results can be explained by the type of designs used in this study. Orthogonal array designs are primarily used to estimate the main effects only, similar to the supersaturated designs. Further, the orthogonal array designs in this study are of resolution III, which means the main effects are aliased with two-factor interaction effects. This makes it difficult for the algorithm to report the true power for detecting active effects.

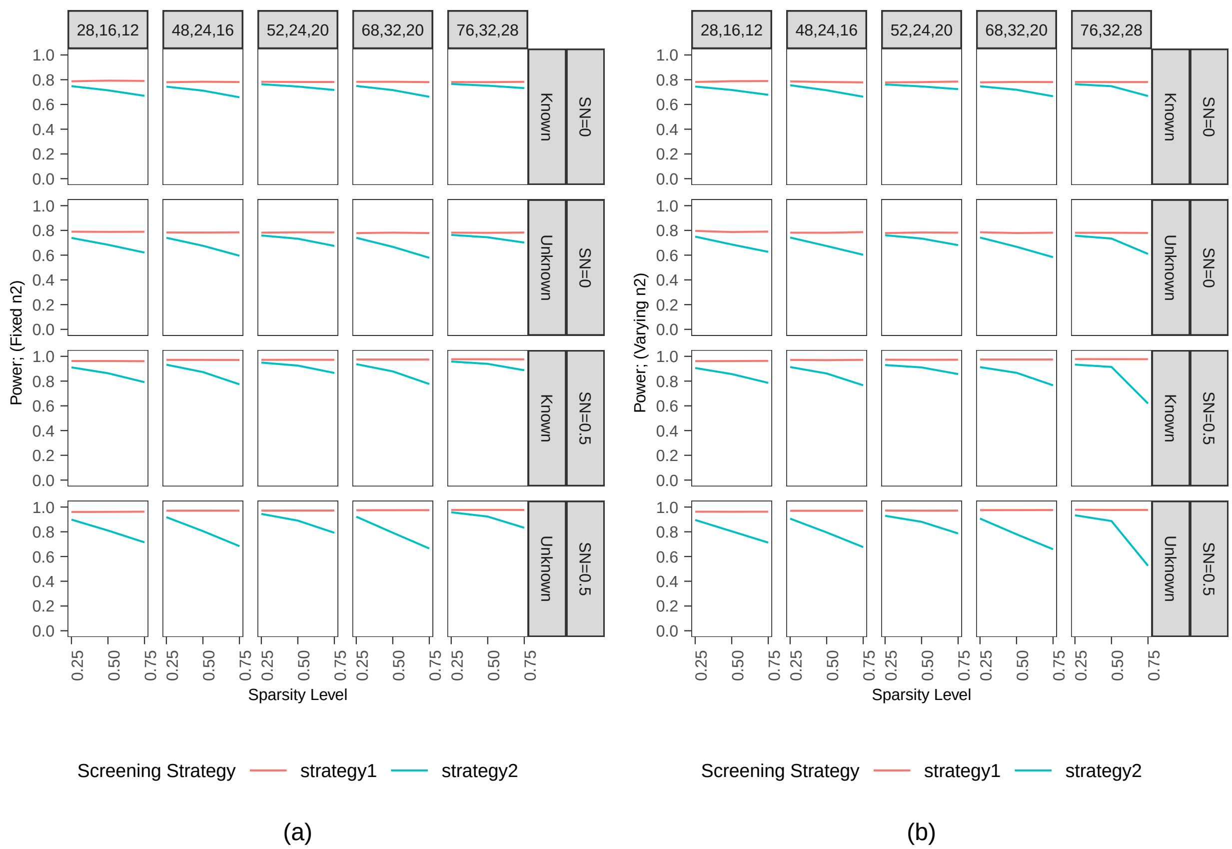

Figure 2: Model 1 power. (a) Power plot when the follow-up design runs are fixed (n2=N-n1); (b) Power plot when the follow-up design runs varies with the active factors found in stage one.

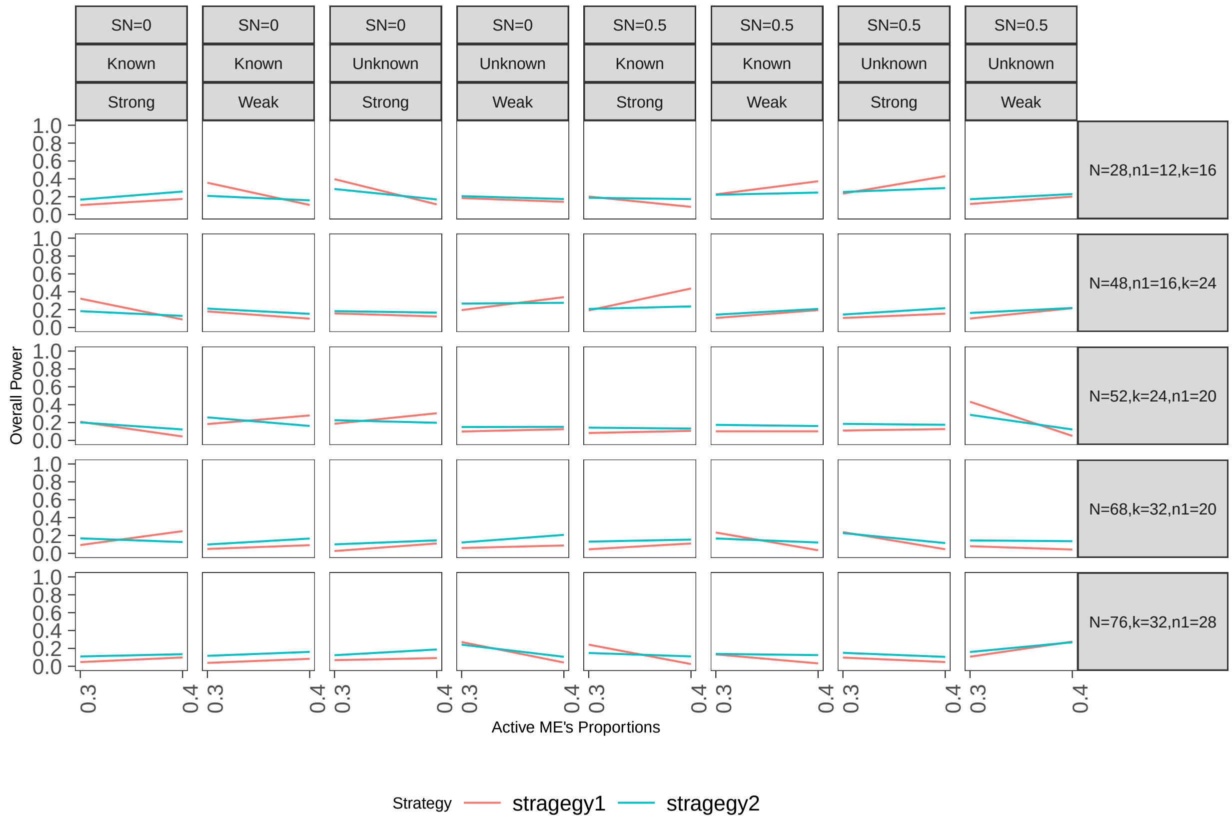

Figure 3: Model 2 overall power.

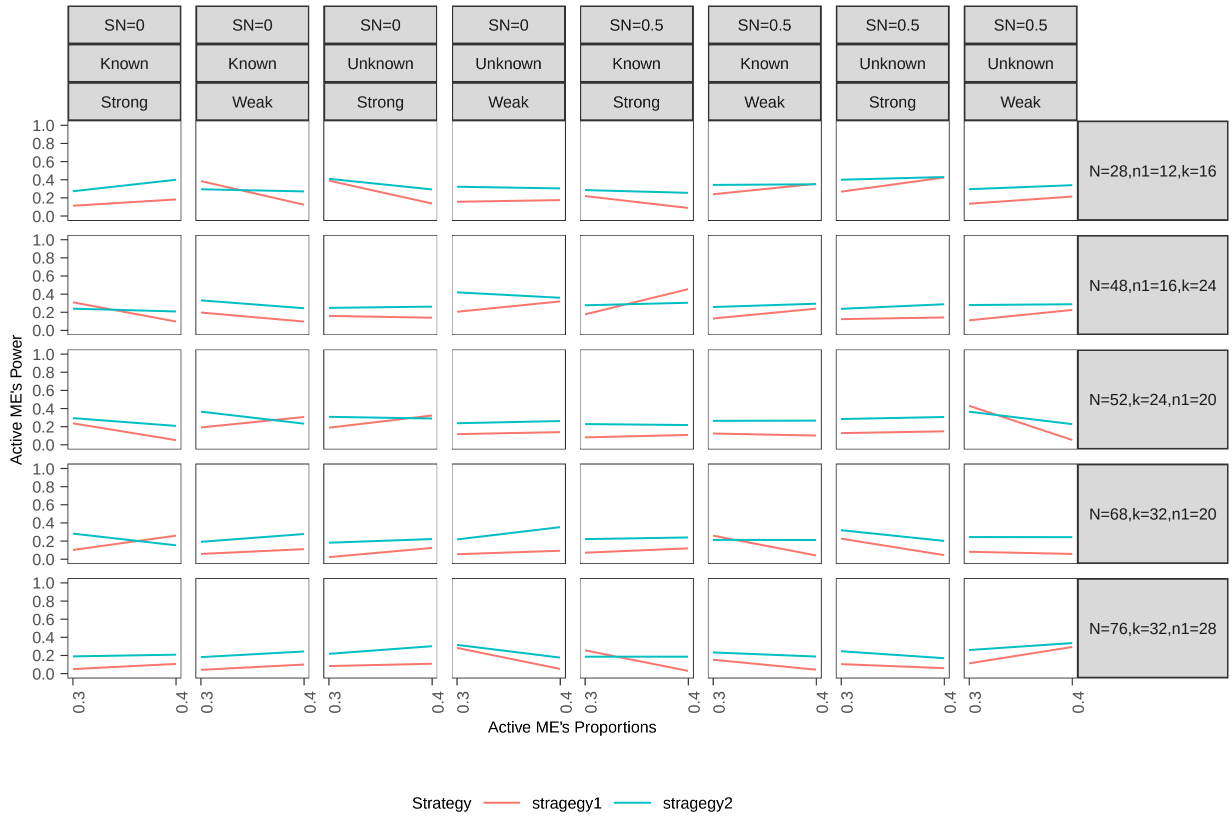

Figure 4: Model 2 main effects power.

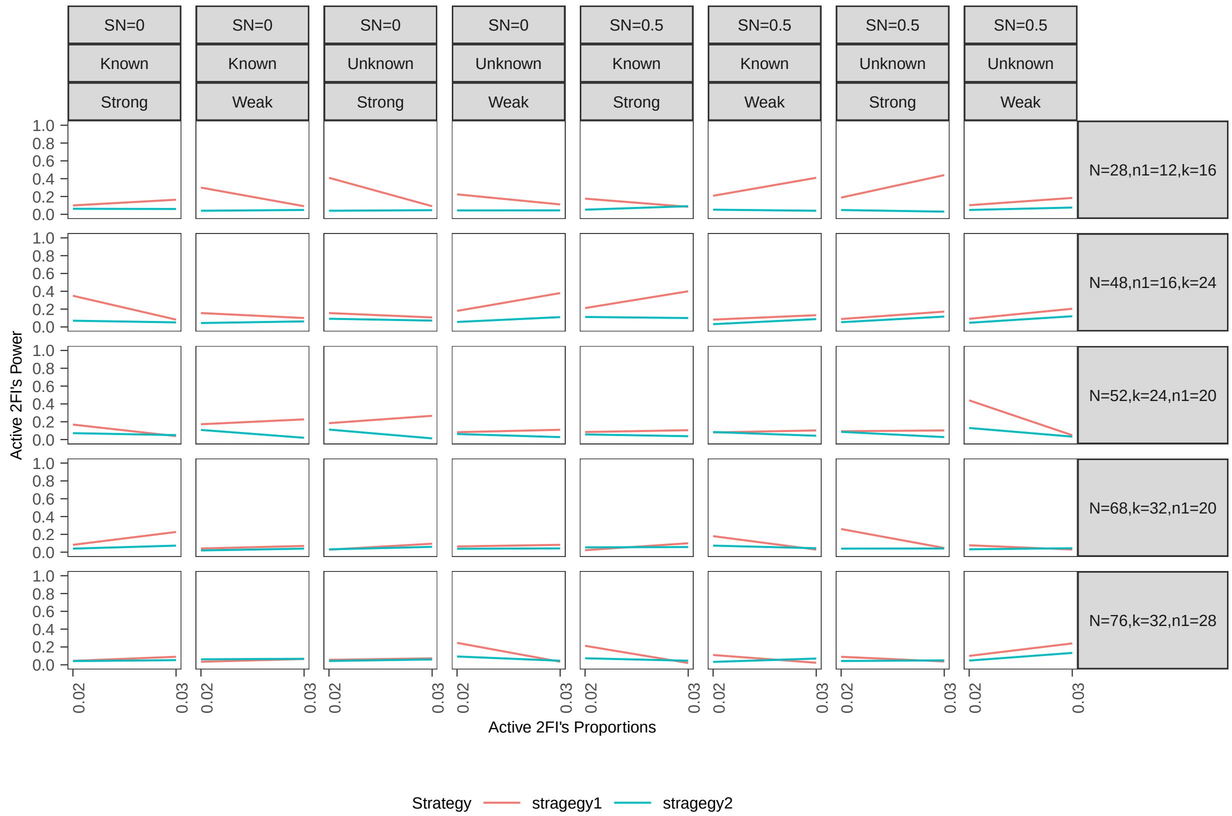

Figure 5: Model 2 two-factor interactions effects power.

3.2 Type I Error Results

Figure 6 shows that strategy 2 has a relatively higher type I error than strategy 1. When the follow-up design runs are allowed to vary, we noticed inflation of the type I error rate for strategy 1. This could be associated with the follow-up designs generated when you allow n_2 to vary. Most of them are unreplicated designs which would increase the variability in the response variable. The two strategies report almost the same type I error rate when two-factor interaction effects are included in a model containing the main effects only (Figures 7, 8, and 9). Although, in some instances, strategy 2 explicitly reports relatively higher type I error.

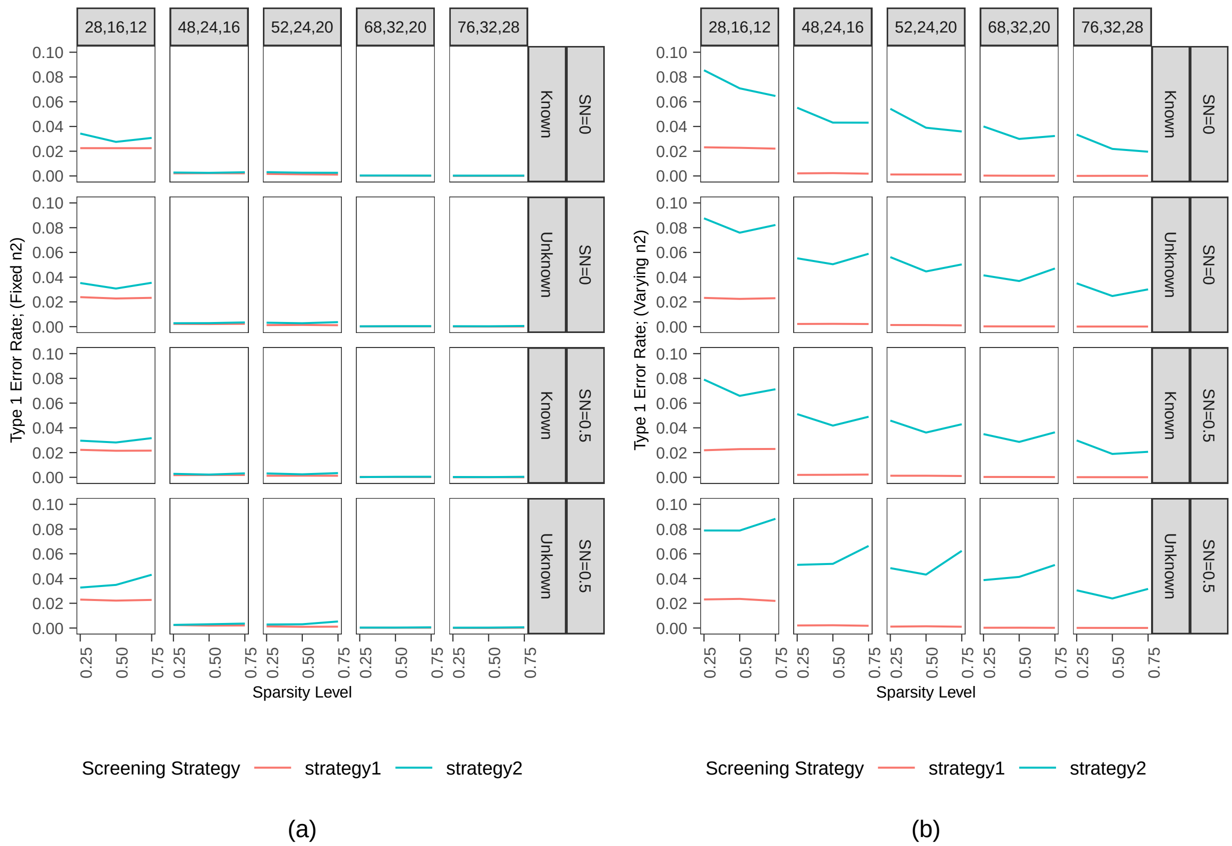

Figure 6: Model 1 Type I Error. (a) Type I error plot when the follow-up design runs are fixed (n2=N-n1); (b) Type I error plot when the follow-up design runs varies with the active factors found in stage one.

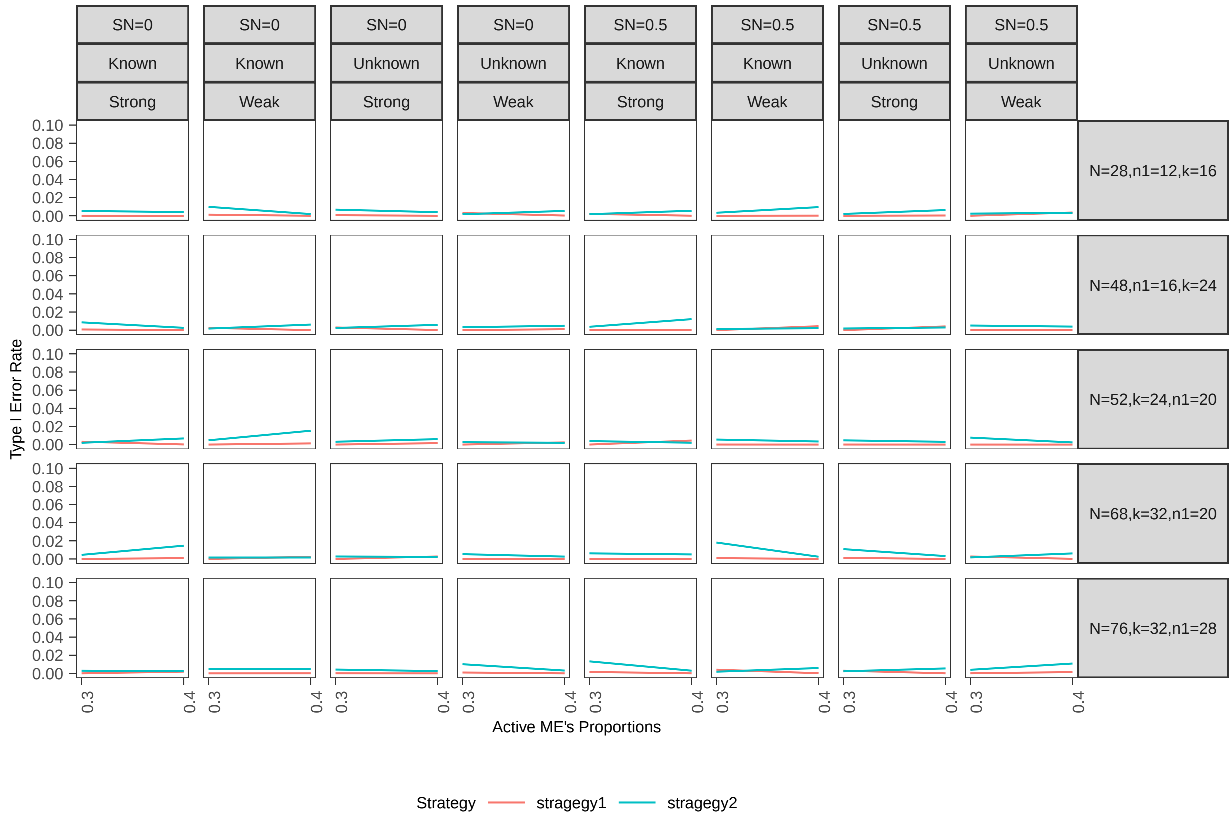

Figure 7: Model 2 overall type I error.

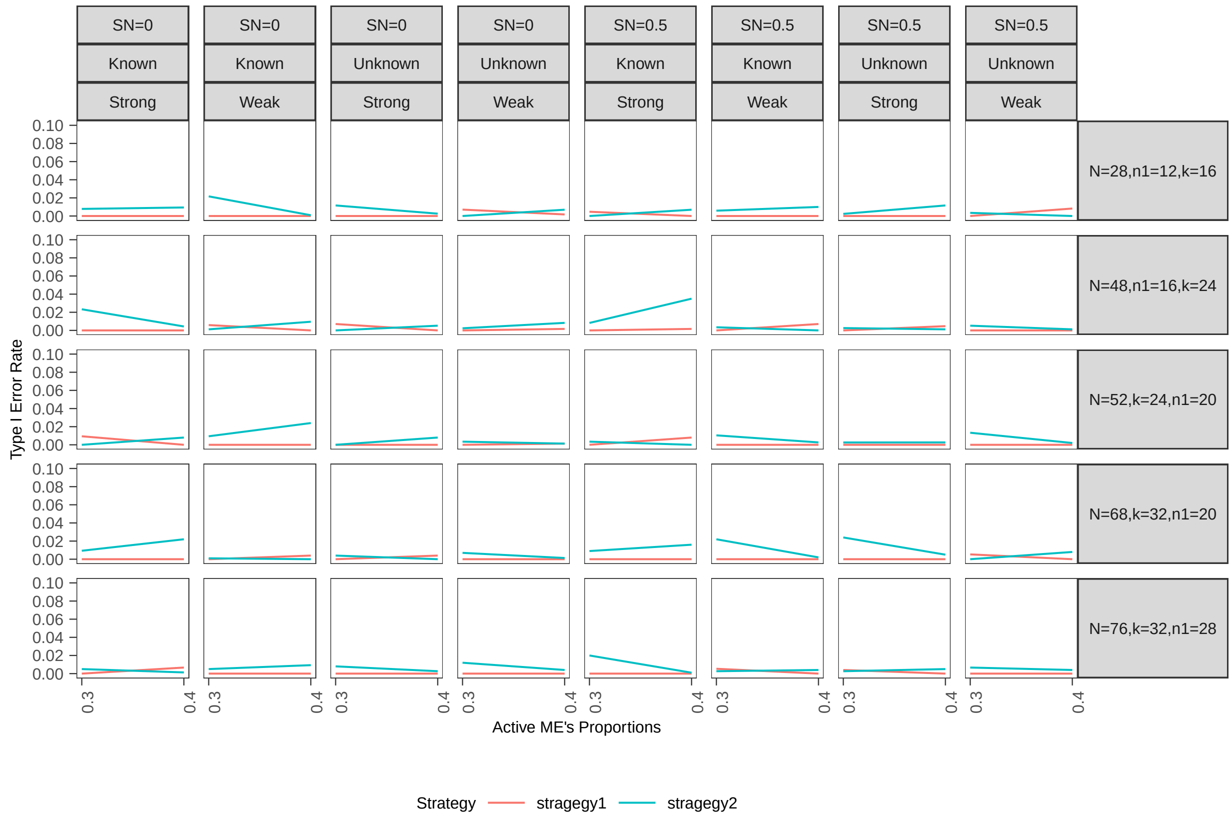

Figure 8: Model 2 main effects type I error.

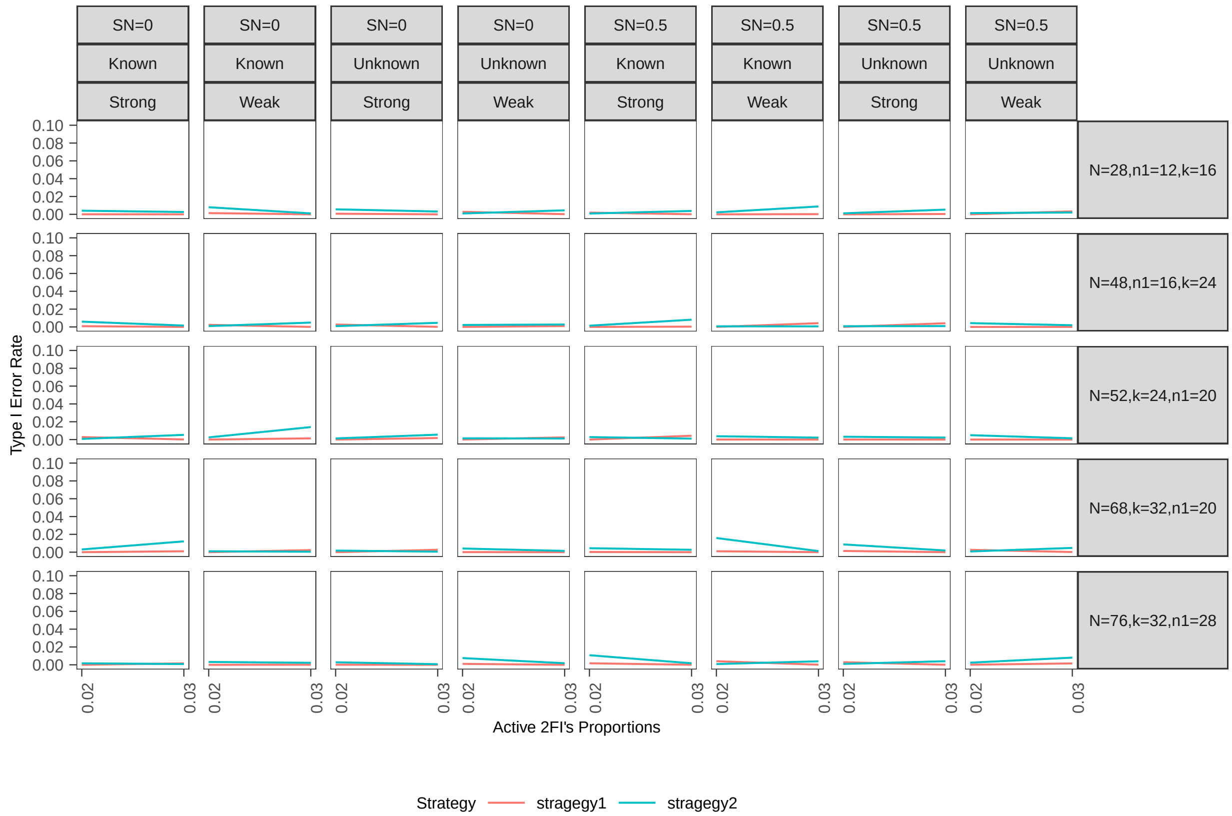

Figure 9: Model 2 two-factor interactions effects type I error.

3.3 False Discovery Rate

Figures 10, 11, 12, and 13 shows that strategy 2 reports more inactive effects as active. This decreases with the decrease in the sparsity level. Allowing the follow-up design runs to vary with the active effects found saves a reasonably good number of runs (Figure 14). However, this inflates strategy 2 false discovery rate.

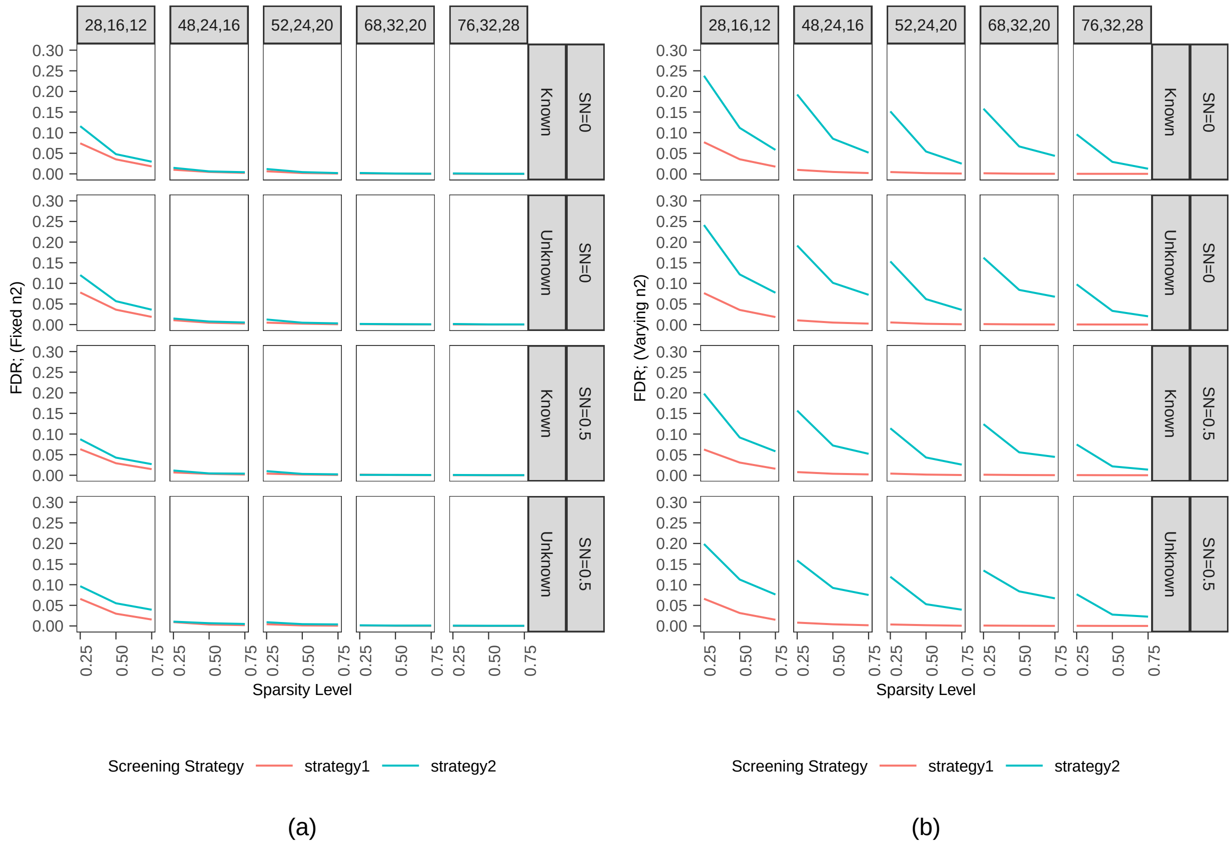

Figure 10: Model 1 false discovery rate. (a) FDR plot when the follow-up design runs are fixed (n2=N-n1); (b) FDR plot when the follow-up design runs varies with the active factors found in stage one.

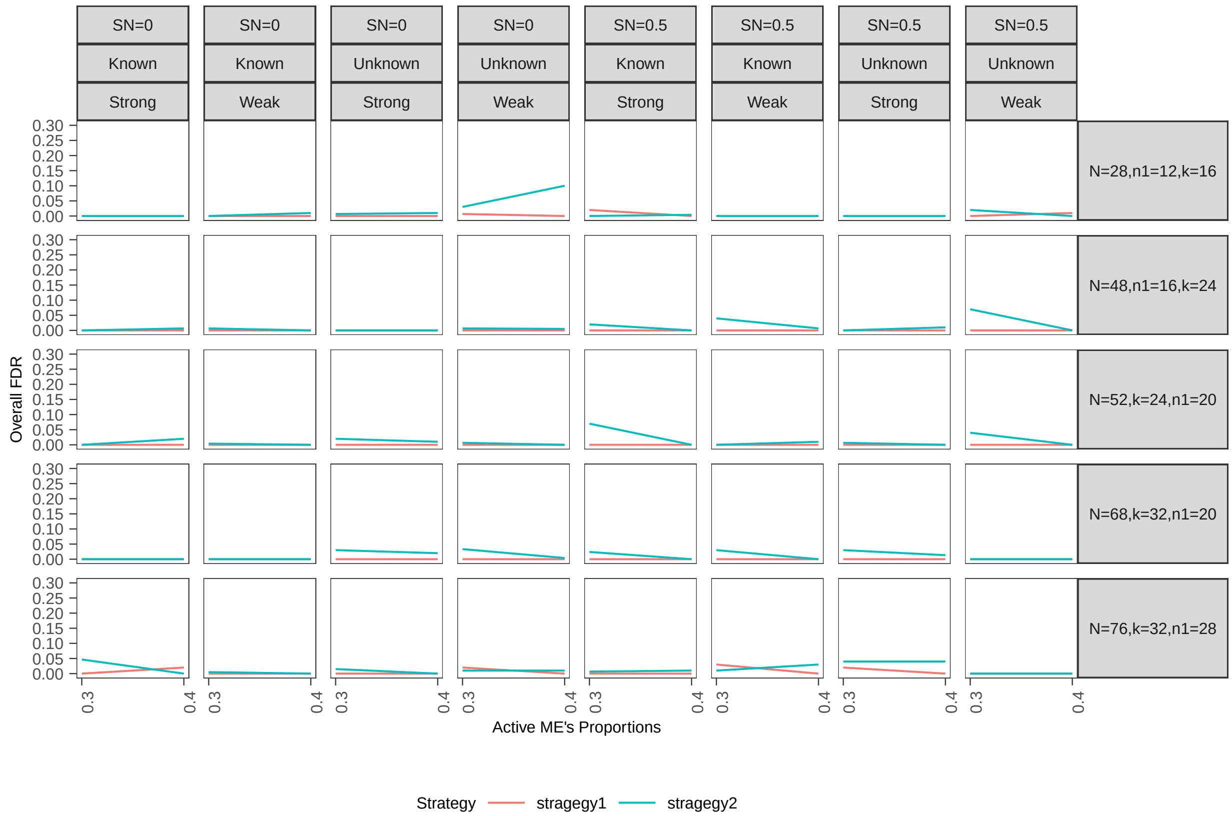

Figure 11: Model 2 overall false discovery rate.

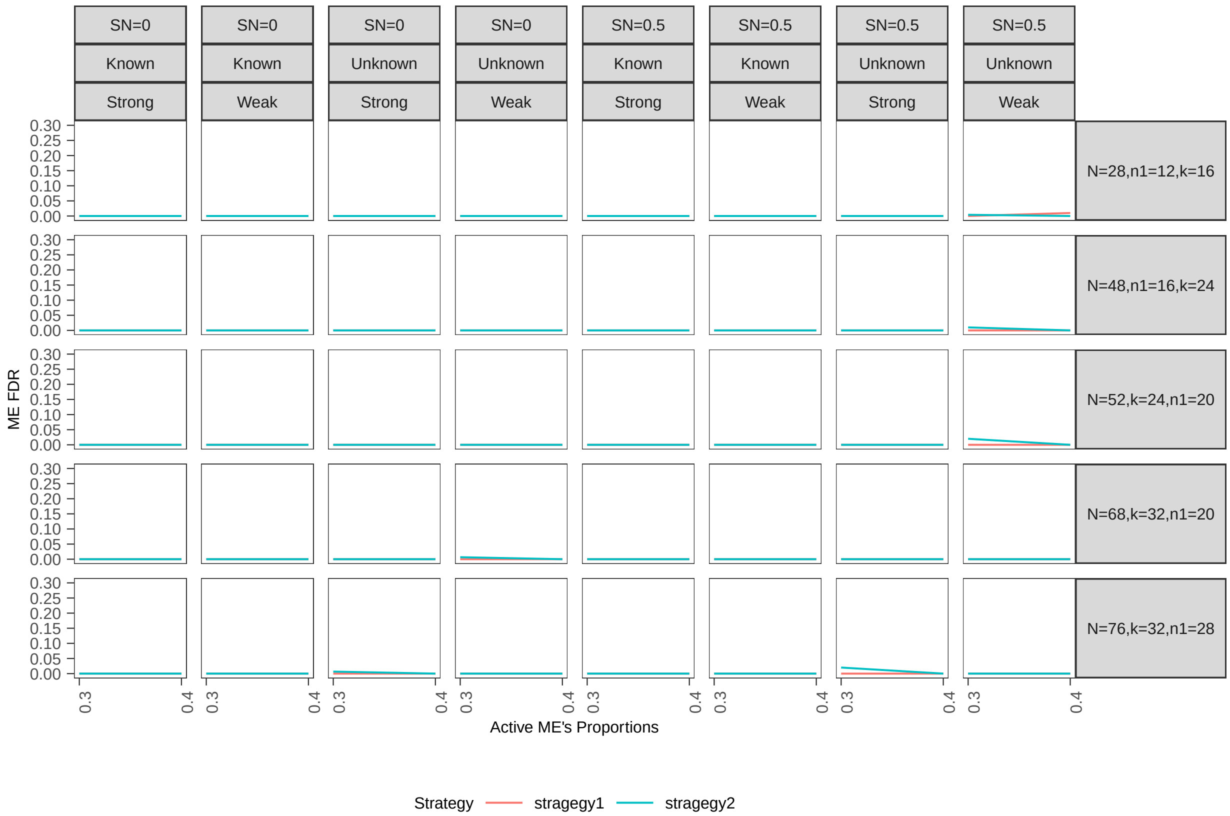

Figure 12: Model 2 main effects false discovery rate.

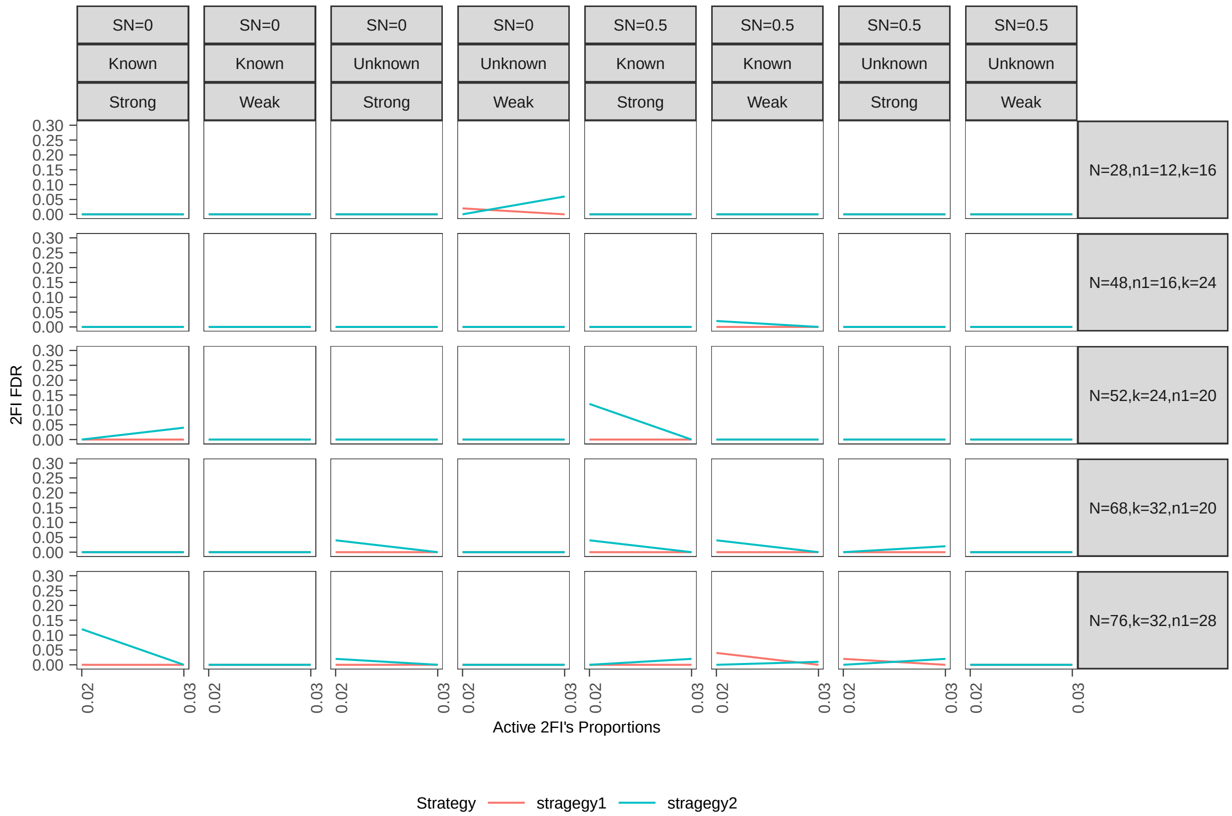

Figure 13: Model 2 two-factor effects false discovery rate.

3.4 Saved Runs

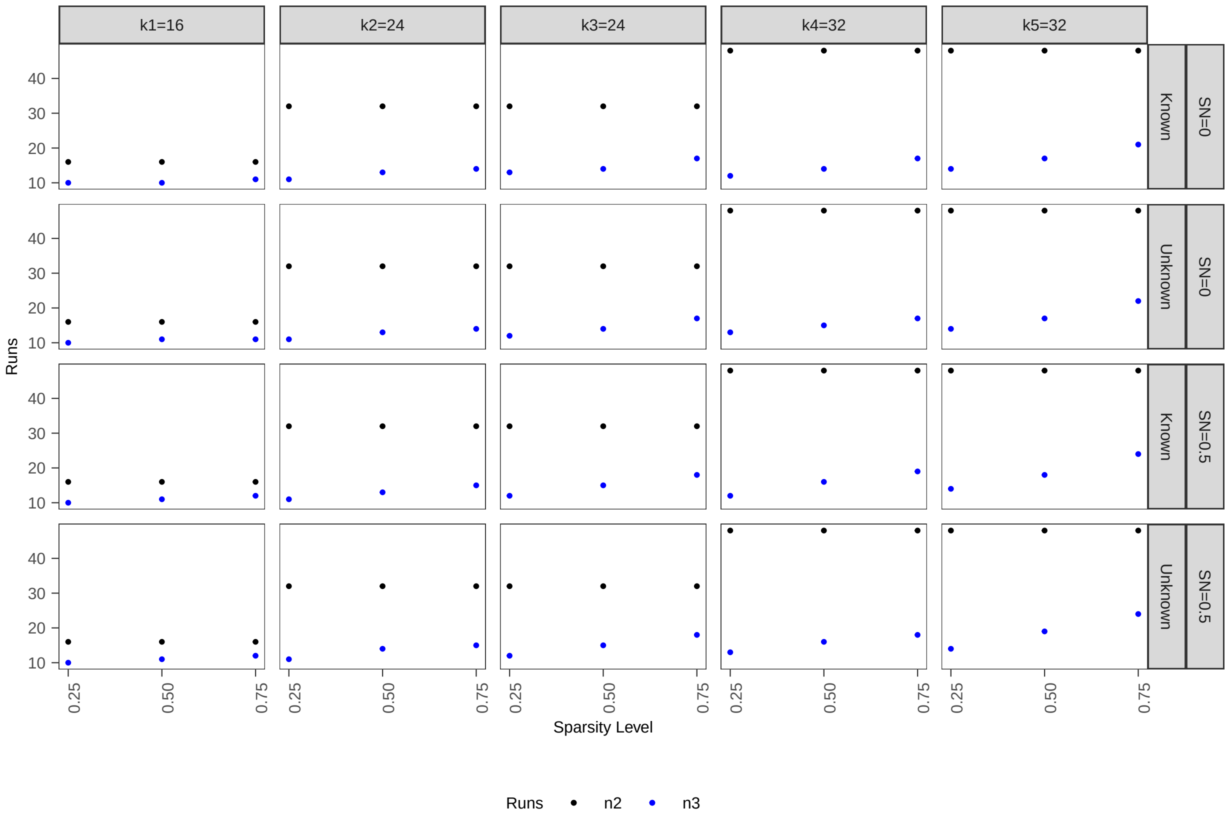

Figure 14: Model 1 saved runs when the follow-up design runs varies with the active factors found in stage one.

4 Discussion and Conclusion

We have investigated different models and scenarios in this study to determine which strategy performs better in screening experiments. While we are willing to conclude that strategy 1 performs better than strategy 2, we can see cases where strategy 2 performs relatively the same with strategy 1: when sparsity is high. Screening experiments are often conducted to identify the most important main effects to engage in the main experiments. Two-factor interactions are rarely investigated in the screening experiment. We extended our study to include two-factor interactions, and we noticed that the performance of the two strategies was relatively poor compared to when only main effects are screened. This may be explained by the type of designs used in our study. The orthogonal array designs in strategy 1 were of resolution III designs, which means the main effects are aliased with the two-factor interactions. This implies that estimating main effects would be equivalent to estimating two-factor interactions aliased to these main effects. In such a scenario, detecting the active main effects and two-factor interactions would be ambiguous (it is unclear to tell which effect [main or two-factor interaction] is causing the change in the system). On the other hand, supersaturated designs are not suitably designed to estimate interactions.

We conclude that using a single-shot experiment with an orthogonal array design would better screen out the least important effects. The experimenter should consider a high sparsity level when using the two-stage screening strategy (SSD + OA follow-up) for better results. We recommend using the difference between OA runs and SSD runs as the follow-up design runs. This would avoid inflation of the type I error rate and the false discovery rate as witnessed in Figures 6 (b) and 10 (b), respectively. However, suppose the experimenter is willing to allow some active factors to be classified as inactive while saving some runs. In that case, one can allow the follow-up design runs to vary with the active effects found in stage one of the analysis. If the experimenter can specify the direction of the effects in advance and not shift the mean of the coefficients, better results could be guaranteed where the effect direction is known. Misspecification of the direction of the effects would not significantly affect the results, as reported by Weese, Edwards, and Smucker (2017).

5 Future Research Work

One of the limitations in this study was scant literature on supersaturated designs follow-up approaches. There are a few other follow-up approaches that would be interesting to investigate their performance. One, Gutman et al. (2014), suggests a follow-up approach based on Bayesian $D-$optimality (Gutman et al. (2014)). Second, a collaborative research *Powerful Regularization-Based Screening Experimentation for Process Optimization with Applications to Additive Manufacturing* suggested augmenting SSDs as a follow-up approach, although there is scant literature on augmentation.

As mentioned earlier, resolution III designs do not do an excellent job in estimating the two-factor interactions. Using resolution IV (strength 3) orthogonal would be something to find out how it would compare with our results. Testing our algorithms on a real data example would also be something interesting. The example would confirm if our results are a true reflection of a real experiment.

Appendix

Search for reasonable thresholds

Table 4: Grid search to find reasonable thresholds

Direction	SN	Sparsity Level	Designs Infomation	$0.5 * \hat{\sigma}$	$\hat{\sigma}$	$1.5 * \hat{\sigma}$	$0.1 * Max\lvert\hat{\beta}_j\rvert$	$0.05 * Max\lvert\hat{\beta}_j\rvert$	$0.25 * Max\lvert\hat{\beta}_j\rvert$	$0.5 * Max\lvert\hat{\beta}_j\rvert$
0.0	0.0	0.25	N=28,n1=12,k=16	0.4907749	0.9815497	1.472325	0.3533552	0.1766776	0.8833881	1.766776
0.5	0.0	0.25	N=28,n1=12,k=16	0.4936114	0.9872228	1.480834	0.4054413	0.2027207	1.0136033	2.027207
0.0	0.5	0.25	N=28,n1=12,k=16	0.4964188	0.9928375	1.489256	0.4515195	0.2257598	1.1287988	2.257598
0.5	0.5	0.25	N=28,n1=12,k=16	0.4958810	0.9917620	1.487643	0.4434192	0.2217096	1.1085481	2.217096
0.0	0.0	0.50	N=28,n1=12,k=16	0.4971232	0.9942464	1.491370	0.5125640	0.2562820	1.2814100	2.562820
0.5	0.0	0.50	N=28,n1=12,k=16	0.4873365	0.9746731	1.462010	0.3542021	0.1771011	0.8855053	1.771011
0.0	0.5	0.50	N=28,n1=12,k=16	0.4951605	0.9903210	1.485481	0.4007086	0.2003543	1.0017716	2.003543
0.5	0.5	0.50	N=28,n1=12,k=16	0.4959387	0.9918774	1.487816	0.4521361	0.2260681	1.1303403	2.260681
0.0	0.0	0.75	N=28,n1=12,k=16	0.4972456	0.9944912	1.491737	0.4380976	0.2190488	1.0952441	2.190488
0.5	0.0	0.75	N=28,n1=12,k=16	0.4979111	0.9958222	1.493733	0.5201416	0.2600708	1.3003540	2.600708
0.0	0.5	0.75	N=28,n1=12,k=16	0.4877675	0.9755349	1.463302	0.4061687	0.2030843	1.0154216	2.030843
0.5	0.5	0.75	N=28,n1=12,k=16	0.4929019	0.9858037	1.478706	0.4503304	0.2251652	1.1258260	2.251652
0.0	0.0	0.25	N=48,n1=16,k=24	0.4942238	0.9884475	1.482671	0.5056138	0.2528069	1.2640346	2.528069
0.5	0.0	0.25	N=48,n1=16,k=24	0.4953989	0.9907977	1.486197	0.4949026	0.2474513	1.2372565	2.474513
0.0	0.5	0.25	N=48,n1=16,k=24	0.4980659	0.9961318	1.494198	0.5674757	0.2837378	1.4186892	2.837378
0.5	0.5	0.25	N=48,n1=16,k=24	0.4891459	0.9782919	1.467438	0.4002329	0.2001164	1.0005822	2.001164
0.0	0.0	0.50	N=48,n1=16,k=24	0.4954884	0.9909768	1.486465	0.4470984	0.2235492	1.1177460	2.235492
0.5	0.0	0.50	N=48,n1=16,k=24	0.4957008	0.9914016	1.487102	0.4972640	0.2486320	1.2431599	2.486320
0.0	0.5	0.50	N=48,n1=16,k=24	0.4983037	0.9966073	1.494911	0.4884286	0.2442143	1.2210715	2.442143
0.5	0.5	0.50	N=48,n1=16,k=24	0.4982892	0.9965784	1.494868	0.5641203	0.2820602	1.4103009	2.820602
0.0	0.0	0.75	N=48,n1=16,k=24	0.4863221	0.9726441	1.458966	0.4783878	0.2391939	1.1959696	2.391939
0.5	0.0	0.75	N=48,n1=16,k=24	0.4966519	0.9933038	1.489956	0.5293854	0.2646927	1.3234636	2.646927
0.0	0.5	0.75	N=48,n1=16,k=24	0.4956524	0.9913047	1.486957	0.5796254	0.2898127	1.4490635	2.898127
0.5	0.5	0.75	N=48,n1=16,k=24	0.4956337	0.9912673	1.486901	0.5676625	0.2838312	1.4191562	2.838312
0.0	0.0	0.25	N=52,k=24,n1=20	0.4964486	0.9928972	1.489346	0.6387504	0.3193752	1.5968760	3.193752
0.5	0.0	0.25	N=52,k=24,n1=20	0.4906843	0.9813686	1.472053	0.4604170	0.2302085	1.1510425	2.302085
0.0	0.5	0.25	N=52,k=24,n1=20	0.4938414	0.9876828	1.481524	0.5162377	0.2581189	1.2905943	2.581189
0.5	0.5	0.25	N=52,k=24,n1=20	0.4955793	0.9911586	1.486738	0.5762730	0.2881365	1.4406826	2.881365
0.0	0.0	0.50	N=52,k=24,n1=20	0.4982326	0.9964651	1.494698	0.5492327	0.2746163	1.3730817	2.746164
0.5	0.0	0.50	N=52,k=24,n1=20	0.4963069	0.9926138	1.488921	0.6419272	0.3209636	1.6048179	3.209636
0.0	0.5	0.50	N=52,k=24,n1=20	0.4873282	0.9746564	1.461985	0.5275822	0.2637911	1.3189556	2.637911
0.5	0.5	0.50	N=52,k=24,n1=20	0.4949181	0.9898362	1.484754	0.5721374	0.2860687	1.4303434	2.860687
0.0	0.0	0.75	N=52,k=24,n1=20	0.4958737	0.9917475	1.487621	0.6330679	0.3165339	1.5826696	3.165339
0.5	0.0	0.75	N=52,k=24,n1=20	0.4979438	0.9958877	1.493831	0.6175049	0.3087524	1.5437621	3.087524
0.0	0.5	0.75	N=52,k=24,n1=20	0.4961292	0.9922583	1.488387	0.6991812	0.3495906	1.7479530	3.495906
0.5	0.5	0.75	N=52,k=24,n1=20	0.4873666	0.9747332	1.462100	0.5072244	0.2536122	1.2680611	2.536122
0.0	0.0	0.25	N=68,k=32,n1=20	0.4945316	0.9890633	1.483595	0.5525890	0.2762945	1.3814725	2.762945
0.5	0.0	0.25	N=68,k=32,n1=20	0.4962638	0.9925276	1.488792	0.6222603	0.3111302	1.5556508	3.111302
0.0	0.5	0.25	N=68,k=32,n1=20	0.4977091	0.9954182	1.493127	0.5938334	0.2969167	1.4845836	2.969167
0.5	0.5	0.25	N=68,k=32,n1=20	0.4981561	0.9963122	1.494468	0.6869767	0.3434883	1.7174416	3.434883
0.0	0.0	0.50	N=68,k=32,n1=20	0.4891877	0.9783753	1.467563	0.5565806	0.2782903	1.3914516	2.782903
0.5	0.0	0.50	N=68,k=32,n1=20	0.4936379	0.9872758	1.480914	0.6043102	0.3021551	1.5107755	3.021551
0.0	0.5	0.50	N=68,k=32,n1=20	0.4950780	0.9901559	1.485234	0.6579389	0.3289695	1.6448473	3.289695
0.5	0.5	0.50	N=68,k=32,n1=20	0.4966202	0.9932405	1.489861	0.6440889	0.3220444	1.6102222	3.220444
0.0	0.0	0.75	N=68,k=32,n1=20	0.4969652	0.9939304	1.490896	0.7209389	0.3604694	1.8023472	3.604694
0.5	0.0	0.75	N=68,k=32,n1=20	0.4886180	0.9772360	1.465854	0.5364027	0.2682013	1.3410067	2.682013
0.0	0.5	0.75	N=68,k=32,n1=20	0.4955302	0.9910605	1.486591	0.5802584	0.2901292	1.4506459	2.901292
0.5	0.5	0.75	N=68,k=32,n1=20	0.4944254	0.9888507	1.483276	0.6477629	0.3238815	1.6194073	3.238815
0.0	0.0	0.25	N=76,k=32,n1=28	0.4962088	0.9924176	1.488627	0.6148204	0.3074102	1.5370509	3.074102
0.5	0.0	0.25	N=76,k=32,n1=28	0.4969155	0.9938309	1.490746	0.7141340	0.3570670	1.7853349	3.570670
0.0	0.5	0.25	N=76,k=32,n1=28	0.4877216	0.9754432	1.463165	0.6087150	0.3043575	1.5217875	3.043575
0.5	0.5	0.25	N=76,k=32,n1=28	0.4924661	0.9849321	1.477398	0.6460260	0.3230130	1.6150649	3.230130
0.0	0.0	0.50	N=76,k=32,n1=28	0.4968914	0.9937827	1.490674	0.7127183	0.3563591	1.7817956	3.563591
0.5	0.0	0.50	N=76,k=32,n1=28	0.4955644	0.9911289	1.486693	0.6931090	0.3465545	1.7327726	3.465545
0.0	0.5	0.50	N=76,k=32,n1=28	0.4969094	0.9938188	1.490728	0.7736797	0.3868399	1.9341994	3.868399
0.5	0.5	0.50	N=76,k=32,n1=28	0.4884689	0.9769378	1.465407	0.5718614	0.2859307	1.4296535	2.859307
0.0	0.0	0.75	N=76,k=32,n1=28	0.4940748	0.9881495	1.482224	0.6246235	0.3123117	1.5615587	3.123118
0.5	0.0	0.75	N=76,k=32,n1=28	0.4945965	0.9891930	1.483790	0.6908427	0.3454214	1.7271068	3.454214
0.0	0.5	0.75	N=76,k=32,n1=28	0.4966960	0.9933921	1.490088	0.6594187	0.3297093	1.6485467	3.297093
0.5	0.5	0.75	N=76,k=32,n1=28	0.4965005	0.9930010	1.489502	0.7651531	0.3825765	1.9128826	3.825765

(i) Fixed n_2: With inactive effects

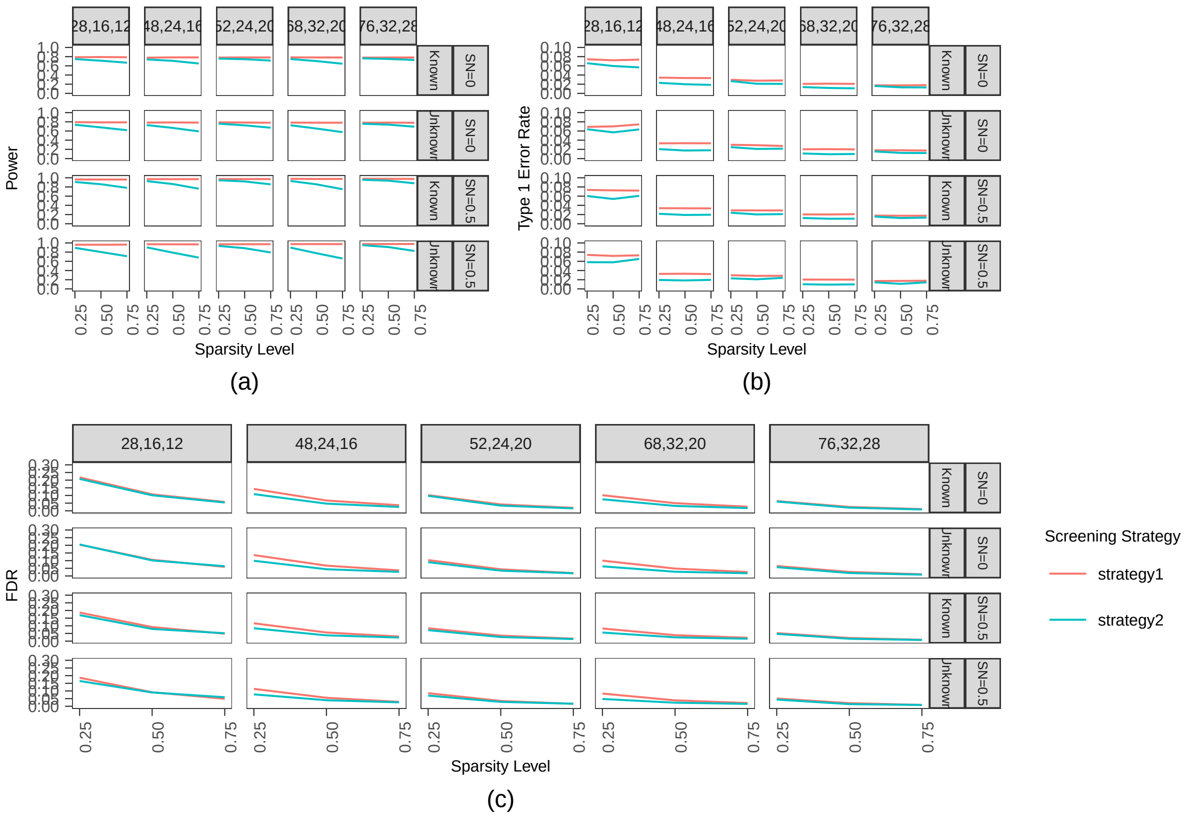

Figure 15: Power, Type I Error and False Discovery Rate (FDR) against sparsity level for model 1 direction (i): (a) shows the power for detecting active effects against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are not varying and including the inactive effects in the model; (b) shows the type I rate against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are not varying and including the inactive effects in the model; (c) shows the false discovery rate against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are not varying and including the inactive effects in the model.

(ii) Varyning n_2: With inactive effects

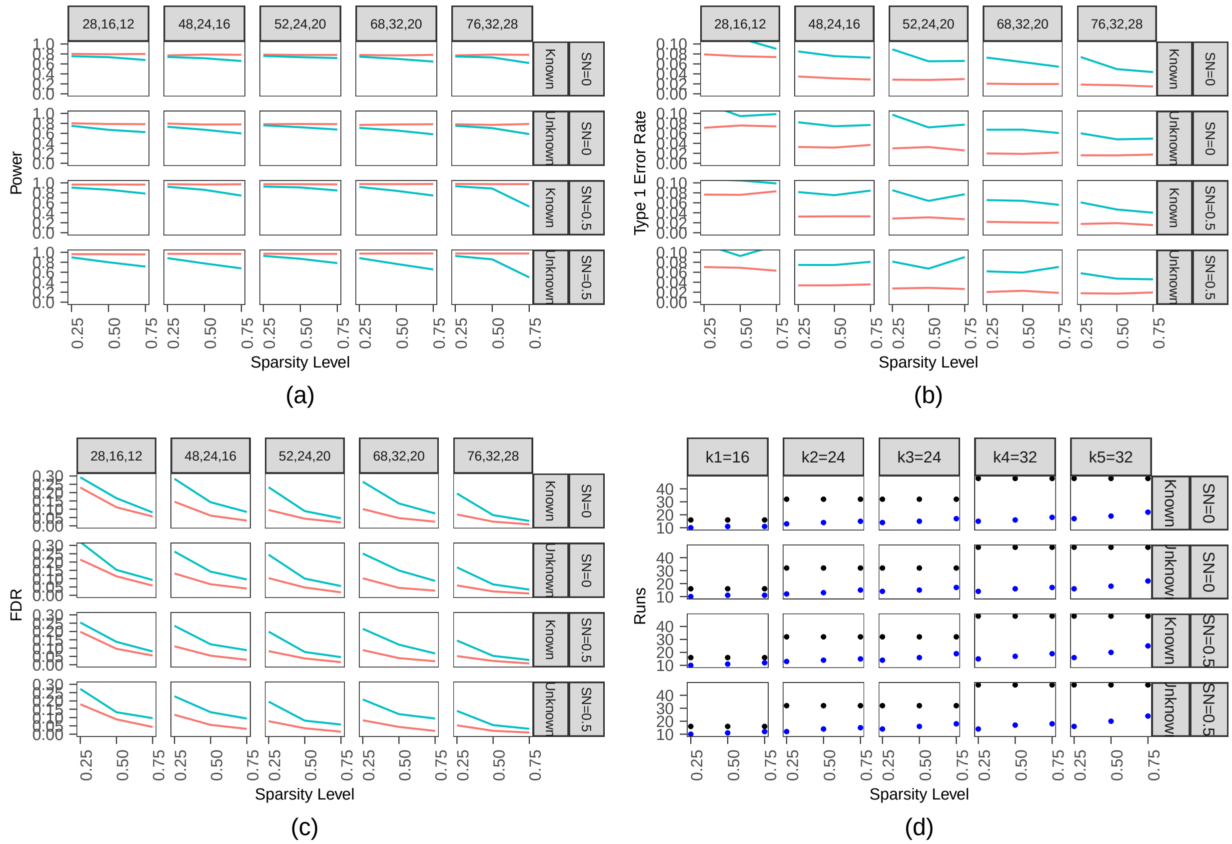

Figure 16: Power, Type I Error and False Discovery Rate (FDR) against sparsity level for model 1 direction (ii): (a) shows the power for detecting active effects against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are varying and inactive effects are included in the model; (b) shows the type I rate against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are varying and inactive effects are included in the model; (c) shows the false discovery rate against sparsity level for OA design and Var(s+)-optimal design + follow-up with OA design, when follow-up design runs are varying and inactive effects are included in the model; (d) shows the comparison of N, n2 and the runs saved when we allow follow-up design runs to vary with the found active factors and when inactive effects are in the model.

References

Booth, Kathleen HV, and David R Cox. 1962. "Some Systematic Supersaturated Designs." *Technometrics* 4 (4): 489–95.

Box, George EP, and R Daniel Meyer. 1986. "An Analysis for Unreplicated Fractional Factorials." *Technometrics* 28 (1): 11–18.

Candes, Emmanuel, and Terence Tao. 2007. "The Dantzig Selector: Statistical Estimation When p Is Much Larger Than n." *The Annals of Statistics* 35 (6): 2313–51.

Dey, Aloke, and Rahul Mukerjee. 2009. "Fractional Factorial Plans" 496.

Grömping, Ulrike. 2018. "R Package DoE. Base for Factorial Experiments." *Journal of Statistical Software* 85 (1): 1–41.

Gutman, Alex J, Edward D White, Dennis KJ Lin, and Raymond R Hill. 2014. "Augmenting Supersaturated Designs with Bayesian d-Optimality." *Computational Statistics & Data Analysis* 71: 1147–58.

Hedayat, A S, NJA Sloane, and J Stufken. 1999. "Orthogonal Arrays: Theory and Applications Springer." *New York*.

Kuhfeld, W. 2010. "Orthogonal Arrays [TS-723]." *Cary (NC): SAS Http://Sup Port. Sas. Com/Techsup/Technote/Ts* 723.

Lin, Dennis KJ. 1993. "A New Class of Supersaturated Designs." *Technometrics* 35 (1): 28–31.

Li, Nandan Sudarsanam, and Daniel D Frey. 2006. "Regularities in Data from Factorial Experiments." *Complexity* 11 (5): 32–45.

Li, and Jeff CF Wu. 1997. "Columnwise-Pairwise Algorithms with Applications to the Construction of Supersaturated Designs." *Technometrics* 39 (2): 171–79.

Marley, Christopher J, and David C Woods. 2010. "A Comparison of Design and Model Selection Methods for Supersaturated Experiments." *Computational Statistics & Data Analysis* 54 (12): 3158–67.

Meyer, Ruth K, and Christopher J Nachtsheim. 1995. "The Coordinate-Exchange Algorithm for Constructing Exact Optimal Experimental Designs." *Technometrics* 37 (1): 60–69.

Phoa, Frederick KH, Yu-Hui Pan, and Hongquan Xu. 2009. "Analysis of Supersaturated Designs via the Dantzig Selector." *Journal of Statistical Planning and Inference* 139 (7): 2362–72.

Rao, C Radhakrishna. 1947. "Factorial Experiments Derivable from Combinatorial Arrangements of Arrays." *Supplement to the Journal of the Royal Statistical Society* 9 (1): 128–39.

Satterthwaite, FE. 1959. "Random Balance Experimentation." *Technometrics* 1 (2): 111–37.

Weese, Maria L, David J Edwards, and Byran J Smucker. 2017. "A Criterion for Constructing Powerful Supersaturated Designs When Effect Directions Are Known." *Journal of Quality Technology* 49 (3): 265–77.

Weese, Maria L, Byran J Smucker, and David J Edwards. 2015. "Searching for Powerful Supersaturated Designs." *Journal of Quality Technology* 47 (1): 66–84.

Weese, Maria L, Jonathan W Stallrich, Byran J Smucker, and David J Edwards. 2020. "Strategies for Supersaturated Screening: Group Orthogonal and Constrained Var (s) Designs." *Technometrics*, 1–13.

Wu, CFJ. 1993. "Construction of Supersaturated Designs Through Partially Aliased Interactions." *Biometrika* 80 (3): 661–69.