Computational Statistics and Data Analysis 194 (2024) 107940

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Factor selection in screening experiments by aggregation over random models

Rakhi Singh [a,*], John Stufken [b]

[a] Binghamton University, New York, USA
[b] George Mason University, Virginia, USA

ARTICLE INFO

Keywords:
Dantzig selector
Main-effects
Screening performance
Supersaturated designs
Two-factor interactions

ABSTRACT

Screening experiments are useful for identifying a small number of truly important factors from a large number of potentially important factors. The Gauss-Dantzig Selector (GDS) is often the preferred analysis method for screening experiments. Just considering main-effects models can result in erroneous conclusions, but including interaction terms, even if restricted to two-factor interactions, increases the number of model terms dramatically and challenges the GDS analysis. A new analysis method, called Gauss-Dantzig Selector Aggregation over Random Models (GDS-ARM), which performs a GDS analysis on multiple models that include only some randomly selected interactions, is proposed. Results from these different analyses are then aggregated to identify the important factors. The proposed method is discussed, the appropriate choices for the tuning parameters are suggested, and the performance of the method is studied on real and simulated data.

1. Introduction

Screening experiments frequently arise in manufacturing engineering. With a large number of potentially important factors, screening experiments are an economical choice for identifying a small number of truly important factors. Such an experiment would then often be followed by a second experiment to study the effects of the selected factors in more detail. We refer the reader to Li et al. (2006) and the references therein for practical applications.

The process to be investigated at the screening stage is typically not well understood, so that one cannot assume that a main-effects model will be adequate. However, with a large number of factors (say 15 or more), there are many model terms to consider if interactions are also included. Based on the effect hierarchy principle, we restrict attention to main-effects and two-factor interactions. In addition, although a qualitative factor may naturally have three or more levels, we restrict consideration to designs with n runs and m two-level factors. With a large number of model terms and a small number of runs, it is common to assume that only a small number of effects are active (effect sparsity). Throughout, we make this assumption.

At the screening stage, the focus is on identifying important factors. Since we only consider main-effects and two-factor interactions, we declare a factor to be important if its main-effect is active or if it is involved in an active two-factor interaction. The effects deemed active by the model selection method are used to select factors for a follow-up experiment.

* Corresponding author at: Department of Mathematics and Statistics, Binghamton University, 400 Vestal Pkwy E, Binghamton, NY 13902, USA.
E-mail address: rsingh@binghamton.edu (R. Singh).

https://doi.org/10.1016/j.csda.2024.107940
Received 29 June 2023; Received in revised form 9 February 2024; Accepted 12 February 2024

Available online 24 February 2024

Any analysis method that considers only the main-effects is bound to make errors if there are active interactions. On the other hand, a method that considers all two-factor interactions at once may have difficulty in identifying active effects, especially with a small number of runs. With m two-level factors, there are $m + \binom{m}{2}$ effects in the model. For $m = 15$, say, there are already 120 terms. With $n = 20$ observations, say, identifying the few active effects among these 120 effects is a very complex problem. We will see that a new method of analysis is needed for problems with this level of complexity.

We propose a new analysis method called *GDS-ARM* (Gauss-Dantzig Selector–Aggregation over Random Models). We apply the Gauss-Dantzig Selctor (GDS) many times, each time with all main-effects and a randomly selected set of two-factor interactions. We then use a method of aggregation over the models suggested by the GDS applications to select the potentially active effects. GDS-ARM gives consideration to two-factor interactions, while reducing the complexity faced by the application of GDS when all two-factor interactions are considered simultaneously. After discussing some preliminaries in Section 2, in Section 3, we suggest a modification to tuning a parameter for GDS which, as we will demonstrate, works well across simulated and real datasets. We introduce the GDS-ARM method in Section 4 for two-level designs along with guidance for the choices for tuning parameters. Note that our method can be easily extended to factors with more than two levels. An assessment of the performance of GDS-ARM will be provided in Section 5 followed by conclusions in Section 6.

2. Preliminaries

For a main-effects model, a two-level design is supersaturated if $n < 1 + m$. The literature on optimality criteria and design construction for such supersaturated designs is vast (see Booth and Cox, 1962; Jones et al., 2008; Jones and Majumdar, 2014; Shi and Tang, 2019; Jones et al., 2020; Weese et al., 2021; Singh and Stufken, 2023; Stallrich et al., 2023, for example). A two-level design is supersaturated for a model with main-effects and two-factor interactions if $n < 1 + m + \binom{m}{2}$. A few examples of such designs that are not supersaturated for main-effects models can be found in Draguljić et al. (2014) who use Bayesian D-optimal designs for $m = 10$, 15, and 20 factors, with $n = 32$, 58, and 94 runs, respectively. These designs perform well even with multiple active two-factor interactions, but can be expensive due to the large values for n. Note that the saturation of any design is defined with respect to the model under consideration. Traditionally, the term "supersaturated design" is used for designs which are supersaturated under the main-effects model. Since we are interested in the model with main-effects and two-factor interactions, we call a design supersaturated if $n < 1 + m + \binom{m}{2}$.

Multiple analysis methods have been studied for screening experiments, including forward selection (Westfall et al., 1998), LASSO (Tibshirani, 1996), smoothly clipped absolute deviation, SCAD (Fan and Li, 2001), the Gauss-Dantzig Selector, GDS (Candès and Tao, 2007; Phoa et al., 2009a), simulated annealing model search, SAMS (Wolters and Bingham, 2011), and hierarchical LASSO (Bien et al., 2013). Marley and Woods (2010) and subsequently, Weese et al. (2015, 2017) showed that GDS has a superior screening performance for a main-effects model, among forward selection, GDS, and model averaging. GDS has also been identified as the best screening method among SCAD, LASSO, GDS, SAMS, Bayesian model selection, and Bayesian model averaging in Draguljić et al. (2014) for a model with main-effects and two-factor interactions. Group screening methods (Draguljić et al., 2014; Jones et al., 2020) or sophisticated integer-programming based methods (Vazquez et al., 2020) are useful, but they typically require a large number of runs or prior knowledge about potentially active effects. Two additional analysis methods are the frequentist approach of Hamada and Wu (1992) involving half-normal plots and forward regression and the Bayesian approach of Box and Meyer (1993). However, these methods are computationally expensive for larger values of m. Recently, McGrath et al. (2023) showed that SCAD should be a preferred choice for models with main-effects and two-factor interactions. Considering all findings listed above, we develop our method based on GDS. While other methods could have been used in principle, a separate study would be needed for comparison and tuning method-specific parameters.

2.1. The model and the Gauss-Dantzig selector

A common model for screening experiments is

$$y = X\beta + \epsilon, \tag{1}$$

where y and ϵ are $n \times 1$ vectors of responses and errors, respectively, X is the $n \times p$ model matrix, and β the $p \times 1$ vector of parameters. For two-level factors, the entries of X would all be ± 1. But we center y and center and scale the columns of X to length $\sqrt{n}$, calling the resulting vector and matrix again y and X, respectively. Note that if a column of X contains the same number of $1s$ and $-1s$, centering and scaling does not change that column. As a result, we do not need an intercept parameter in (1), and take $p = m$ or $p = m + \binom{m}{2}$ for a main-effects model and a model that also includes all two-factor interactions, respectively.

The GDS was first proposed by Candès and Tao (2007) in the multiple regression context. It starts with the Dantzig selector, which obtains the estimator $\hat{\beta}(\delta)$ as a solution to

$$\min_{\beta \in \mathcal{R}^p} ||\beta||_1 \quad \text{subject to} \quad ||X^T(y - X\beta)||_\infty \leq \delta, \tag{2}$$

where $||\beta||_1 = |\beta_1| + \cdots + |\beta_p|$ is the l_1 norm, and $||a||_\infty = \max(|a_1|, \ldots, |a_p|)$ is the l_∞ norm. As a second step, only the effects with estimates exceeding γ in magnitude are retained, and are re-estimated using ordinary least squares. The second step is helpful in reducing the bias of the estimates. The whole process is repeated for multiple values of $\delta \in [0, ||X^T y||_\infty]$, leading to multiple models. Model selection criteria like AIC, BIC, adjusted R^2, etc. can then be used to select one of these models (see Phoa et al. (2009a) and

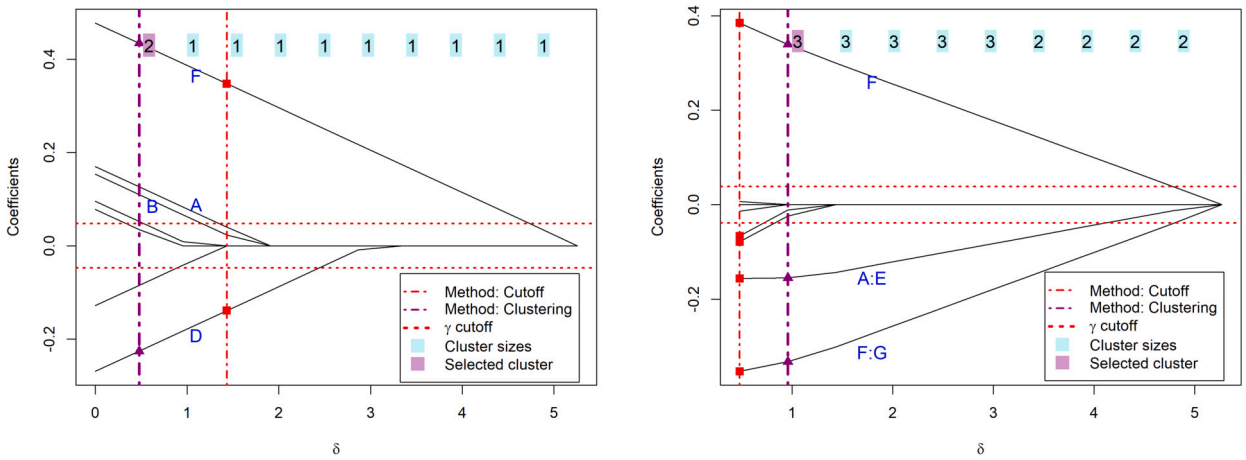

Fig. 1. Profile plots for the cast fatigue example of Hunter et al. (1982) when GDS(m) (left panel) and GDS(m+2fi) (right panel) are applied. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

Marley and Woods (2010)). There is no consensus on how to select a value for γ. Phoa et al. (2009a) used $\gamma = 1$ (corresponding to 10% and 6.7% of the true β in their simulations), while Marley and Woods (2010) use $\gamma = 1.5$. Recently, Weese et al. (2021) evaluate three different choices for γ, and suggest using the data-driven value of $0.1 \times ||\hat{\beta}(0)||_\infty$, where $\hat{\beta}(0)$ is the solution to (2) for $\delta = 0$. A value of γ that is too small results in models that are too large, but if γ is too large we may miss active effects. While we concur that a data-driven selection method is preferred, we have seen too many examples where $0.1 \times ||\hat{\beta}(0)||_\infty$ is not a good choice. Note that the GDS can be applied using only the m main-effects or using the m main-effects and all $\binom{m}{2}$ two-factor interactions. We will refer to these as GDS(m) and GDS(m+2fi), respectively.

3. A cluster-based alternative for tuning γ

Since the data-driven choice of $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ can work poorly, we use an alternative procedure for selecting active effects throughout the paper:

1. For selected values of δ, obtain the Dantzig selector estimate, $\hat{\beta}(\delta)$, of β.
2. For each $\hat{\beta}(\delta)$, apply k-means clustering (Lloyd, 1982) with two clusters on the absolute values of the estimates in $\hat{\beta}(\delta)$, and, using ordinary least squares, refit a model that only contains effects corresponding to the cluster with the larger mean.
3. Select the value of δ that corresponds to the model from the previous step that has the smallest BIC value. The effects in that model are declared to be the active effects.

This method does not explicitly select a value for γ, but has the same effect of shrinking smaller estimates to 0. For δ, we use the 100 equally spaced values $(i/101) \times ||X^T y||_\infty$, $i = 1, ..., 100$. We demonstrate the usefulness of our clustering method by the following example.

Example 1. A cast fatigue experiment with 12 runs and 7 factors was originally studied by Hunter et al. (1982), and was later revisited by Hamada and Wu (1992) and Phoa et al. (2009a), among others. For ease of reference, the data are provided in the Supplementary Material. The seven factors, denoted by capital letters A through G, represent initial structure, bead size, pressure treatment, heat treatment, cooling rate, polish, and final treatment, respectively, whereas the response is the casting lifetime. It is widely accepted for these data that the main-effect of F, also written as F, and the interaction effect of F and G, written as FG, are active effects, with the AE interaction possibly being active as well (see Hamada and Wu (1992)).

Fig. 1 shows the profile plots when GDS(m) (left panel) and GDS(m+2fi) (right panel) are applied to these data. The profile plots show how the Dantzig selector effect estimates change as a function of the tuning parameter δ. The red horizontal lines denote the $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ cutoff, and effects with estimates beyond the two red lines will be declared active when using GDS with this choice of γ. The numbers in the top row show the number of effects declared active by our method (step 2 at the beginning of this section) for a given choice of δ. For ease of reading, the figure only displays these numbers for every tenth δ value. The vertical red lines show the selected δ value and the corresponding active effects after the application of BIC when the $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ cutoff is used, whereas the vertical magenta lines show the same for the clustering method. Different methods lead to different competing models for the same value of δ. As a result, models with the smallest BIC differ, and ultimately, different models are selected for different methods.

For GDS(m), the left panel of Fig. 1 shows that the main-effects F and D are declared active by both the clustering method and the $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ method. However, GDS(m+2fi), the right panel shows that the clustering method selects F, FG and AE, while the $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ method declares all of F, FG, AE, D, AD, and EF as active. Both methods find factors A, E, F and G to be important,

but factor D is only added to the list by the $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ method. The choice of $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ is simply to small for these data. Also, among F, FG and AE, the latter estimate has the smallest size. For those who believe in weak heredity, AE might not be seen as a real effect, and only factors F and G would be declared as important. ■

The cutoff of $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ was investigated and suggested for the main-effects model. We looked at alternatives for studying models that also include interactions, as with GDS(m+2fi) in Example 1, and selected the clustering method defined in the current section. Based on our experience, it works well on many simulated and real datasets, including in Example 1. It is worth noting that in all our simulations, we consider all active effect sizes to be of the same order of magnitude. If this is not the case, then it is possible that our method can be improved by using more than two clusters. In view of the effect sparsity assumption, three clusters might suffice, with the effects in the two clusters with the largest means being considered as active.

4. The GDS-ARM method

In this section, we describe the GDS-ARM method in more detail and recommend choices for its tuning parameters. Its performance will be studied in Section 5. But first, we motivate our method.

4.1. Motivation

Any screening method can fail to identify one or more of the important factors or declare one or more unimportant factors erroneously as important. GDS(m), which only considers main-effects, can be effective if there are no active interactions. But it often fails to identify a factor that is only active through an interaction. Moreover, if an active interaction is highly correlated with a main-effect, GDS(m) may incorrectly identify this main-effect as active. This is precisely what happens in Example 1, where the main-effect D is highly correlated with the FG interaction. With $p = 7 + 21$ effects and 12 runs, GDS(m+2fi) works well for this example, declaring F, FG and AE as active, which is widely accepted to be the correct answer for these data. However, with a small number of runs n, the performance of GDS(m+2fi) rapidly declines with increasing values of the number of factors m. The results from the simulated data in Section 5 will confirm that both GDS(m) and GDS(m+2fi) perform poorly under those conditions.

Instead, our method, GDS-ARM, applies GDS many times, each time with all main-effects and a randomly selected set of two-factor interactions. We then use a method of aggregation over the models obtained from the multiple GDS applications to select the potentially active effects. As a final step, we apply a stepwise regression starting from the model with the selected effects, and giving main-effects that were not selected an opportunity to enter the model. We call this method *GDS-ARM* (Gauss-Dantzig Selector–Aggregation over Random Models). GDS-ARM reduces the complexity by each time using only some of the two-factor interactions, akin to random forest (Breiman, 1996; Friedman et al., 2001).

4.2. The algorithm

Schematically, GDS-ARM consists of four steps:

(a) Apply GDS for multiple models that contain all main-effects and randomly selected interaction columns.
(b) Using a model selection criterion, identify the top models from those obtained in part (a).
(c) Select effects by aggregating over the selected top models in part (b).
(d) Apply stepwise regression starting with a model that contains the effects selected in part (c), and give main-effects that were left out an opportunity to enter the final model.

Effects that appear in the model selected in step (d) are declared to be active, and any factor that is part of at least one such effect is declared to be important.

Algorithm 1: GDS-ARM.

inputs: the number of applications *nrep*, the number of selected two-factor interactions *nint*, the number of top models *ntop*, the minimum proportion *pkeep* of top models in which an effect should appear to be selected, a design d, and the response vector y

for $j = 1 \rightarrow nrep$ **do**
  Let $X_{m,nint}$ be a $n \times (m + nint)$ matrix with all rows of X, all m main-effect columns, and *nint* randomly selected interaction columns;
  Apply the GDS procedure in Section 3 using y and $X_{m,nint}$;
  Save the selected active effects in the list B_j after deleting interactions that violate the weak heredity principle;
end
Using BIC, find the best *ntop* models (represented by $\{B_{(1)}, \ldots, B_{(ntop)}\}$, say) from the *nrep* models corresponding to $B_1, \ldots, B_{nrep}$;
Select the effects from $\{B_{(1)}, \ldots, B_{(ntop)}\}$ that appear in at least $\lceil pkeep \times ntop \rceil$ of these lists;
Starting with the model that contains all selected effects, perform stepwise regression on selected effects and all m main-effects;
After deleting interactions that violate the weak heredity principle, effects selected by the stepwise regression are declared to be active ;
Declare a factor to be important if it appears in at least one active effect;
output: important factors from step 10

Table 1
Parameter tuning for GDS-ARM.

Param.	Description	Levels	Values	Selected value
nrep	Number of GDS applications	3	$\frac{1}{2}\binom{m}{2}, \binom{m}{2}, 2\binom{m}{2}$	$\binom{m}{2}$
nint	Number of 2fi in each GDS application	2	$0.2\binom{m}{2}$, $0.4\binom{m}{2}$	$0.2\binom{m}{2}$
ntop	Number of top models	2	$\max\left(20, \frac{nrep \times nint}{2\binom{m}{2}}\right)$, $\frac{2nrep \times nint}{\binom{m}{2}}$	$\max\left(20, \frac{nrep \times nint}{2\binom{m}{2}}\right)$
pkeep	Proportion of the *ntop* models in which an effect should appear to be selected	3	0.10, 0.25, 0.4	0.25

For step (a) (lines 1–5 in Algorithm 1), GDS is performed *nrep* times as described in Section 3, each time with *nint* randomly selected two-factor interactions. Thus, the model matrix X will each time be an $n \times (m + nint)$ matrix, say $X_{m,nint}$, with m main-effects columns and *nint* interaction columns. One could try to select the set of interaction columns more judiciously, for example by using balanced incomplete block designs (see Smucker and Drew (2015), who use a representative sample of models using balanced incomplete block designs to generate optimal designs), but randomly selected columns appear to work well and impose no restrictions on the values for *nrep* and *nint*. In addition, in line 4 of the algorithm, we impose weak heredity (Hamada and Wu, 1992; Chipman et al., 1997; Yuan et al., 2007, 2009). We modify result of each GDS model by deleting active interaction effects for which neither of the corresponding main-effects is active in line 4 before saving the active effects While there are other ways in which weak heredity, or even strong heredity, can be enforced, our investigations suggest that the method proposed here is most effective for identifying the important factors if weak or strong heredity holds. We will briefly return to this at the end of Section 5.

In step (b) (line 6 in Algorithm 1), we select *ntop* models using the BIC criterion following Marley and Woods (2010) and Weese et al. (2021). Other choices are possible, such as modified AIC (Phoa et al., 2009a) or corrected AIC (Dragulјić et al., 2014), however, these choices often result in very small models (see, for example Marley and Woods, 2010).

Step (c) (line 7 in Algorithm 1) discards effects appearing in less than $pkeep \times ntop$ of the top models, where *pkeep* is the proportion of top models in which an effect needs to appear in order to be selected. This eliminates effects that appear, perhaps by chance, in only very few top models.

Finally, with step (d) (line 8 of Algorithm 1), incorrectly selected effects can exit the model, while main-effects have another chance to enter the model. This preferential treatment for main-effects aligns with the effect hierarchy principle. In addition, after the stepwise model selection, we retain only those interactions that are consistent with the weak heredity principle. In principle, one could use GDS instead of stepwise regression, but stepwise gives the opportunity to start with and improve a model that contains effects that appear to be active based on the earlier steps.

The basic idea and motivation for GDS-ARM comes from random forest. With random forest, instead of using just one tree, multiple trees are used, just as we use multiple GDS applications. And instead of using all variables to build a tree, only randomly selected variables are used each time. As a result, GDS-ARM facilitates inclusion of two-factor interactions while avoiding the complexity that GDS(m+2fi) would face by considering all these interactions simultaneously.

4.3. Parameter tuning

Aside from the GDS parameters δ and γ (see Section 3), there are four tuning parameters for Algorithm 1: *nrep*, *nint*, *ntop*, *pkeep*.

For each of the *nrep* GDS applications, we use *nint* two-factor interactions. Since there are $\binom{m}{2}$ such interactions, there are $\binom{\binom{m}{2}}{nint}$ different models that we could fit. This number will almost always be too large. Moreover, while *nrep* should not be taken too small, we find that there is a point of diminishing returns. In fact, the number of unimportant factors that are wrongly declared important becomes unacceptably large if *nrep* is too large. For *nint*, note that GDS(m) and GDS(m+2fi) correspond to $nint = 0$ and $nint = \binom{m}{2}$, respectively. We find that a much smaller value for *nint* than $\binom{m}{2}$ is preferable, although the best choice can depend on (1) the number of runs n and (2) the number of active two-factor interactions. With larger values for these numbers (albeit that the number in (2) can only be guessed, such as by using some guidance from Li et al. (2006)), one could consider taking *nint* larger than our recommendation.

The number of top models, *ntop*, is bounded by *nrep*. However, some of the fitted models may not provide a good fit because the randomly selected interactions for that GDS application did not include one or more active interactions. On average, an interaction is selected in $nrep \times nint / \binom{m}{2}$ GDS applications. To give active interactions a good chance to show up in the top models, we would want to take *ntop* smaller than this number. Finally, for an effect not to be discarded at this stage, we would want it to appear in at least $100 \times pkeep$% of the top models. Effects that appear in very few of the top models may have appeared there by chance. On the other hand, we would not want *pkeep* to be too large, because active interactions could appear in only a modest number of the top models.

We performed a simulation study using a complete factorial for the four tuning parameters with the 36 level combinations shown in Table 1. We studied screening performance across simulation scenarios S1-S7 described in Subsection 5.2, using 100 iterations for each level combination and for one design for each of (n, m) = (12, 16), (18, 22), and (24,16). Detailed results can be found in the Supplementary Material. For each of the seven scenarios, we perform an analysis of variance with main-effects and two-

factor interactions, using the difference between the true positive rate and false positive rate as the response. This resulted in the recommendations provided in the final column of Table 1.

Thresholds for the p-values in the stepwise procedures in steps 8 and 9 of Algorithm 1 are additional tuning parameters. Throughout our real and simulated studies, we set the threshold for adding a term at 0.01 and for removing a term at 0.05.

While these recommendations for the tuning parameters work extremely well in virtually all situations that we considered, as with any other method that includes tuning parameters, sometimes other choices can give better results.

5. Screening performance

We apply GDS-ARM to two real datasets and to various simulated datasets. In the simulations, we will compare GDS-ARM to other screening methods by using four commonly used measures:

1. True Positive Rate (TPR): The probability that important factors are correctly identified;
2. False Positive Rate (FPR): The probability that unimportant factors are declared being important;
3. True Factor Identification Rate (TFIR): The rate at which all important factors are identified correctly as important;
4. Size: The total number of factors identified as important.

These metrics depend not only on the method, but also on the design, the number of active effects, and how many of the active effects are main-effects. In a simulation study, for a given method, design, and setting, the TPR and FPR can be approximated by proportions based on multiple iterations. For simplicity, we will continue to refer to these proportions as TPR and FPR. The TFIR and Size are approximated by taking average TFIR and average Size values over multiple iterations.

Increasing TPR tends to increase FPR and TFIR. A method should have high TPR because missing an important factor at the screening stage is highly undesirable. But if the FPR is too large, too many unimportant factors will be kept for a follow-up experiment. With a slightly higher emphasis placed on higher TPR than on smaller FPR, we consider a method to be good if it has higher TPR and reasonably small FPR. In addition, Size should be close to the total number of important factors, and TFIR should be as high as possible. As we will see, GDS(m) performs well in terms of the four metrics if there are no active interactions. We will also see that, irrespective of the presence of active interactions, the performance of GDS(m+2fi) can be poor, especially if n is relatively small. The poor performance of GDS(m+2fi) can primarily be attributed to the fact that GDS needs to consider quite a few columns with some of them being highly correlated. For example, for $n = 16, m = 24$, GDS(m+2fi) has the sheer impossible task of considering $24 + \binom{24}{2} = 300$ highly correlated potential effects in just 16 runs. Throughout this section, for every GDS application, we use the method provided in Section 3. We also compare our method to LASSO, SCAD, and SAMS. But first, we see the results of GDS-ARM on the real case studies.

5.1. Real case studies

Example 1 revisited. First, we apply GDS-ARM, with values for the tuning parameters as in the last column of Table 1, to the experiment from Example 1. For the cast fatigue experiment with 12 runs, 7 factors and 28 effects, we saw that GDS(m) incorrectly identifies factors D and F as important, whereas GDS(m+2fi) identifies factors A, E, F and G (Example 1). The application of GDS-ARM with $nrep = 21$, $nint = 5$, $ntop = 20$ and $pkeep = 0.25$ (from Table 1) identifies F and FG as the active effects, and thus F and G as the important factors. This agrees with the conclusion in Hamada and Wu (1992), who would not add the AE interaction to this model based on the heredity principle. Others, like Westfall et al. (1998) and Phoa et al. (2009a), discuss the possibility that the AE interaction is also active. In Example 4, we will introduce a modified GDS-ARM method that does not insist on satisfying weak heredity. With that method, the conclusion would have been that F, FG and AE are the active effects and, as with GDS(m+2fi), that A, E, F and G are the important factors. ■

Example 2. An analytical experiment was conducted by Dopico-García et al. (2007) to characterize the chemical composition of white "Vinho Verde" grapes, simultaneously determining the most important phenolic compounds and organic acids for the grapes. We follow Phoa et al. (2009b) by considering one phenolic compound, kaempferol-3-Orutinoside + isorhamnetin-3-O glucoside. The experiment used a 12-run Placket-Burman design with eight factors (A–H). The data is provided in the Supplementary Material for the sake of completeness. Fitting a main-effects model suggests that factors D and F are important (the corresponding model has $R^2 = 41\%$). Upon reanalyzing the data, Phoa et al. (2009b) found that the active effects are C, D, and AD ($R^2 = 93\%$), so that the important factors are A, C, and D. They also note that the factor F was misidentified in the main-effects analysis perhaps due to its partial aliasing with the interaction AD. From Table 2, we see that both GDS(m) and GDS(m+2fi) misidentify important factors, whereas GDS-ARM with parameters as in Table 1 reaches the same conclusion as in Phoa et al. (2009b). ■

5.2. Simulated datasets

For given numbers of n runs and m factors, we consider seven scenarios, say S1 through S7. The numbers of active main-effects, c_1, and two-factor interactions, c_2, are taken as $(c_1, c_2) = (3,0), (4,0), (5,0), (3,1), (4,1), (3,2)$ and $(4,2)$ for S1 through S7, respectively. Unless otherwise stated, two-factor interactions can only be active if at least one of the corresponding main-effects is active (weak heredity). Hence, there are at most $c_1 + c_2$ important factors. GDS-ARM is expected to do even better compared to GDS(m) if there

Table 2
GDS results on the chemical composition experiment in Example 2.

Method	Parameters	Imp. factors
GDS(m)	BIC, Clustering	B, D, E, F
GDS(m+2fi)	BIC, Clustering	A, B, D, E, G, H
GDS-ARM	$nrep = 28, nint = 6,$ $ntop = 20, pkeep = 0.25$	A, C, D

are active interactions between factors whose main-effects are not active, even though such situations are not considered here. So, scenarios S1-S3 have 0 active two-factor interactions (which favors GDS(m)), while the other scenarios have 1 or 2 active two-factor interactions. All coefficients for active effects (including the intercept) are generated from $N(5,1)$, with a randomly selected sign. Coefficients corresponding to inactive effects are set to 0. Errors are generated as independent $N(0,1)$. We also present results for the mean effect sizes equal to 3 and 1.5. Using a design d, and corresponding model matrix X (prior to centering and standardizing X), the response is then generated using the model in Equation (1). For each scenario and design, a response vector is generated 1000 times. Average screening performance over the 1000 iterations is measured by TPR, FPR, TFIR, and Size.

In Examples 3 and 4, we present results for $(n,m) = (18,22)$ and $(24,16)$, respectively, and show that GDS-ARM is superior to the other methods in our comparisons. In Example 5, we consider performances of designs with varying $n = 14 - 20$ and $m = 24$. These combinations of (n,m) have been used in previous studies and designs used have different properties (for example, some are $E(s^2)$-optimal, some are $UE(s^2)$-optimal, etc.). Details about the optimality properties of these designs have been provided in the Supplementary Material. While we studied multiple designs for each case, since performance is similar, results for only one design in each case are presented, with those for other designs being relegated to the Supplementary Material. For GDS(m) and GDS(m+2fi), we use BIC and the clustering-based method for parameter tuning. While use of the adaptive value of $\gamma = 0.1 \times ||\hat{\beta}(0)||_\infty$ would have resulted in slightly larger TPR, it would also have resulted in considerably larger FPR. For GDS-ARM we use the parameters suggested in Table 1 and the methodology described in Section 3. Both LASSO and SCAD are applied with all main-effects and two-factor interaction columns, with BIC being used to select an appropriate tuning parameter. SAMS is applied with the default values as suggested in Wolters and Bingham (2011) and used in McGrath et al. (2023).

Example 3. For $n = 18$ and $m = 22$, we use the $E(s^2)$-optimal design in Marley and Woods (2010). The tuning parameters for GDS-ARM are $nrep = 231$, $nint = 46$, $ntop = 23$, and $pkeep = 0.25$. The mean effect sizes of 5, 3, and 1.5 for the coefficients of active effects are considered. We have ensured that for an active interaction at least one of the factors has an active main-effect. Fig. 2 displays the results for the mean effect size of 5, whereas Fig. 3 does the same for the mean effect sizes of 3 and 1.5. The four panels correspond to the four different metrics.

Fig. 2 shows that both SCAD and GDS(m+2fi) performs poorly in terms of all metrics. GDS(m) performs well for the scenarios without interactions (S1-S3), but falls short for scenarios S4-S7. SAMS selects the smallest model and, as a result, has a small FPR, but also a small TPR, especially for the more complicated scenarios. LASSO selects a large number of factors, which results in a high TPR and unacceptably high FPR. GDS-ARM is competitive with GDS(m) for S1-S3, and superior for S4-S7. The four panels on the left in Fig. 3 show the results for mean effect size 3, whereas those on the right are for mean effect size 1.5. The conclusions for mean effect size 3 are the same as those for mean effect size 5 in Fig. 2. For mean effect size 1.5, all methods are challenged, but GDS-ARM still yields the best TPR among methods with a reasonable FPR. Based on the results in this example, especially SCAD, LASSO, and GDS(m+2fi) have FPRs that are too large to be considered viable competitors.

Three additional designs for $n = 18$, $m = 22$ are considered in the Supplementary Material: two $UE(s^2)$-optimal designs (Jones and Majumdar, 2014), and a Bayes D-optimal design (from Marley and Woods, 2010). GDS-ARM outperforms all other considered methods across scenarios and designs. ■

Example 4. For $n = 24$ and $m = 16$, we use design 20.1 from Schoen et al. (2017). Designs 20.2a, 20.2b, and 20.3a of Schoen et al. (2017) are also studied in the Supplementary Material. Active effects are selected by three different methods, with coefficients randomly chosen as positive or negative and sizes coming from the $N(5,1)$ distribution. The cases for mean effect sizes of 3 and 1.5 are presented in the Supplementary Material. The three different methods for the active effects are: (1) No restriction on which main-effects and interactions are active, i.e., no heredity assumption is imposed for generating the simulated data; (2) strong heredity is imposed for generating the simulated data; and (3) weak heredity is imposed for generating the simulated data (but strong heredity is allowed if the random selection of active effects leads to such a model). The tuning parameters used for GDS-ARM are $nrep = 120$, $nint = 24$, $ntop = 20$, and $pkeep = 0.25$.

In Fig. 4, where data generation does not impose any heredity restriction, among the five methods GDS(m), GDS(m+2fi), SCAD, SAMS, LASSO, we find that GDS(m+2fi) has arguably the best screening performance. LASSO and SCAD continue to select a large number of factors, making the FPR unacceptably high. GDS(m) and SAMS select too few factors, leading to lower TPR values. The latter also holds for GDS-ARM. However, in pink with marker "M", we present a "modified" version of GDS-ARM. Modifications are made in lines 4 and 9 of Algorithm 1 by no longer deleting interactions that violate the weak heredity assumption. This modified GDS-ARM method outperforms GDS(m+2fi) with considerably higher TPR and TFIR values and an FPR that is at most that of GDS(m+2fi). Thus, when data are generated by models that do not impose any heredity properties, then the modified GDS-ARM method is the clear winner.

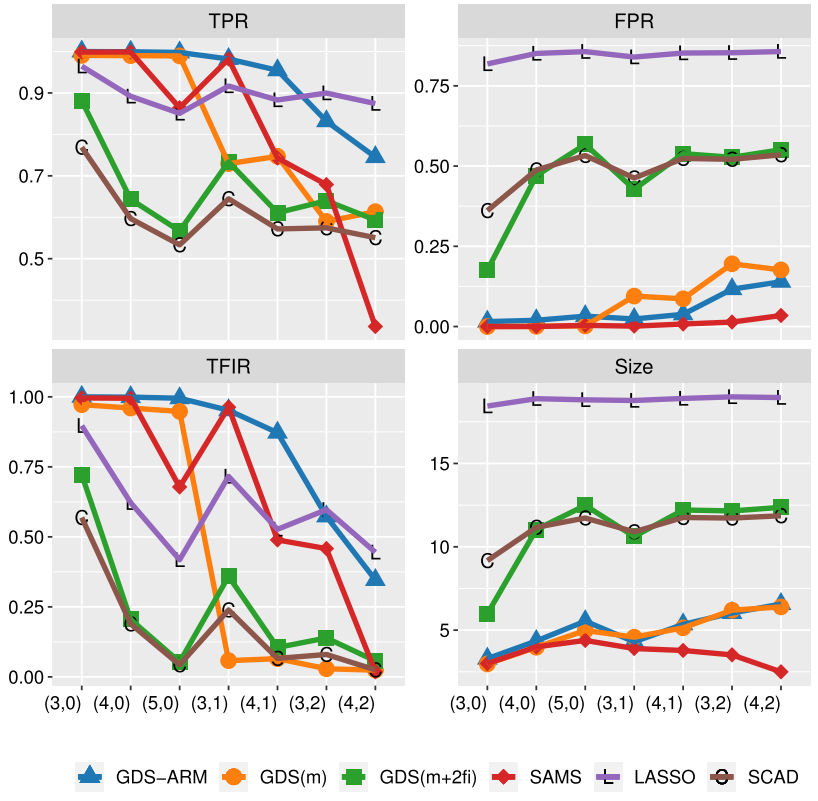

Fig. 2. Four performance metrics over 1000 iterations for $n = 18, m = 22$ for an E(s^2)-optimal design when coefficients of active effects are generated from a normal distribution with standard deviation 1 and mean effect size of 5.

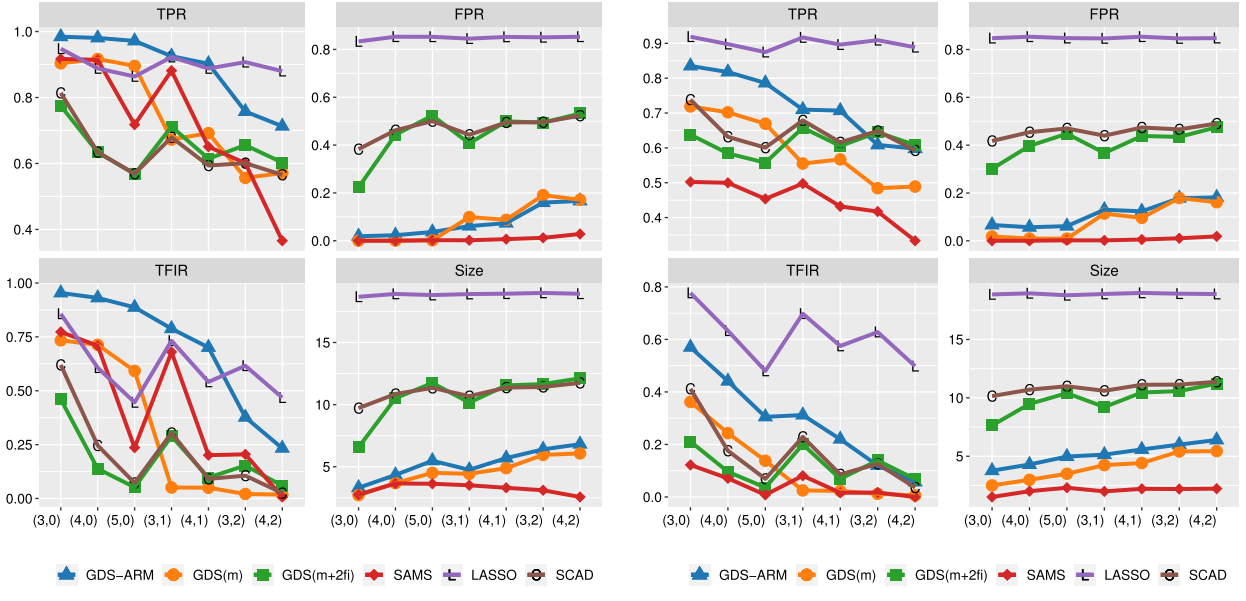

Fig. 3. Four performance metrics over 1000 iterations for $n = 18, m = 22$ for an E(s^2)-optimal design when coefficients of active effects are generated from a normal distribution with standard deviation 1 and mean effect sizes of 3 (four plots on the left) and 1.5 (four plots on the right).

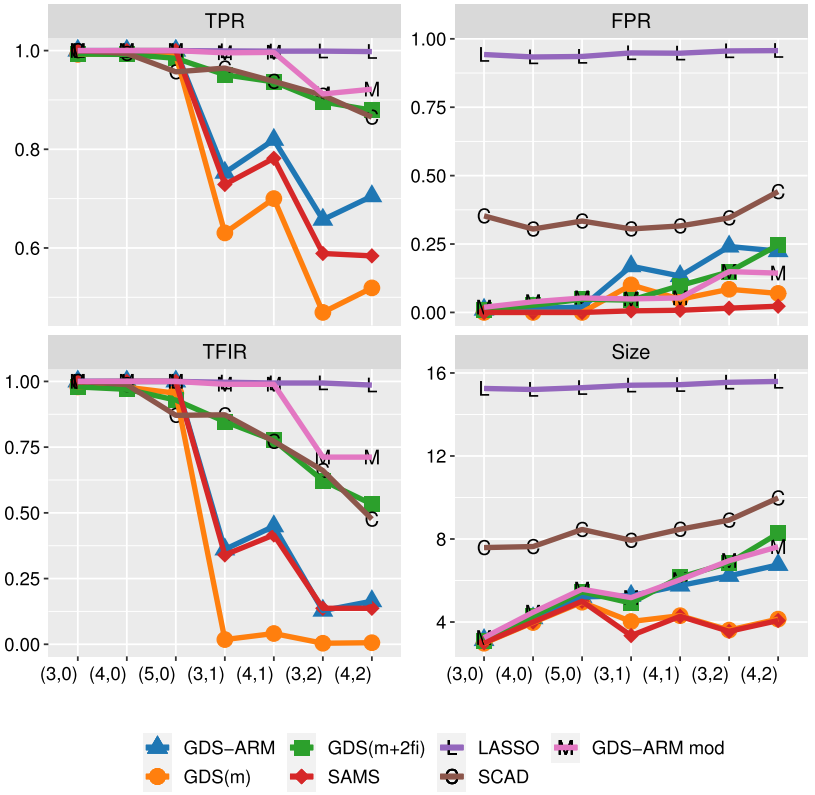

Fig. 4. Four performance metrics over 1000 iterations for $n = 24, m = 16$ when coefficients of active effects are generated from a normal distribution with standard deviation 1 and mean effect size of 5. For an active interaction, there is no restriction on the main-effects, implying that weak heredity may not be satisfied.

Fig. 5 presents the cases where data are generated by models that satisfy either strong heredity (four plots on the left) or weak heredity (four plots on the right). We note that SCAD, LASSO, and GDS(m) perform poorly. GDS(m+2fi) is better in this figure than in Fig. 4, which is, based on our experience, related to n being larger and m being smaller in the current example. Examples for $(n, m) = (20, 16)$ and $(24, 20)$ in the Supplementary Material also demonstrate this feature. However, GDS-ARM and SAMS are the winners in Fig. 5, with SAMS overall even a tad better than GDS-ARM. Fig. 5 also includes results for the modified GDS-ARM method. Since the models that generate the data are now taken to satisfy weak or strong heredity, one might expect that a screening method that ignores the heredity principle, such as the modified GDS-ARM method, would not perform as well. Compared to the GDS-ARM method as proposed in Algorithm 1, the modified version has however approximately the same TPR values, but slightly larger FPR values. Thus, if one is comfortable with the assumption of at least weak heredity, then the GDS-ARM method as proposed in Algorithm 1 is preferred. Results for other designs with $(n, m) = (24, 16)$ are shown in the Supplementary Material when data are generated using the weak heredity assumption. GDS-ARM and SAMS are also the top choices for those designs. We also present results for mean effect sizes 3 and 1.5 for one of the designs under the assumption of weak heredity in the Supplementary Material. While all methods are more challenged, GDS-ARM distinguishes itself, also compared to SAMS, especially in terms of TPR values. ■

Example 5. Keeping m fixed at 24, we now consider four $E(s^2)$-optimal designs for $n = 14, 16, 18$, and 20 respectively. The sizes of coefficients of active effects are again drawn from $N(5, 1)$ and for any active two-factor interaction at least one of the corresponding main-effects must be active. Fig. 6 shows that the performance of all methods improves when n increases. While GDS(m) is competitive with GDS-ARM for $n = 14, m = 24$, GDS-ARM is better in the other three scenarios. We see that for small n, none of the methods performs particularly well. But when n comes closer to m, GDS-ARM is clearly superior. It is not surprising that, with a very small n, it is hard to identify active effects correctly. This latter observation is also visible for the case $(n, m) = (12, 16)$ studied in the Supplementary Material. Since the direction of the results for TFIR and Size do not alter the conclusions, the detailed results for these metrics are relegated to the Supplementary Material.

The values for the tuning parameters for GDS-ARM follow the recommendations in Table 1. The $E(s^2)$-optimal design for $n = 14, m = 24$ is the same as that used in Marley and Woods (2010), whereas designs for other choices of n and m are obtained from https://engineering.purdue.edu/Smartdesigns/twolevel.html. These designs are also provided in the Supplementary Material. ■

The Supplementary Material contains many additional examples that lead to observations that are worth mentioning. First we consider different designs for $(n, m) = (18,22)$, $(12,16)$, and $(24,16)$. For a given choice of n and m, while most designs perform equivalent, there are times when design choice matters such as for (12,16). For $n = 12$ and $m = 16$, we studied five designs, one of

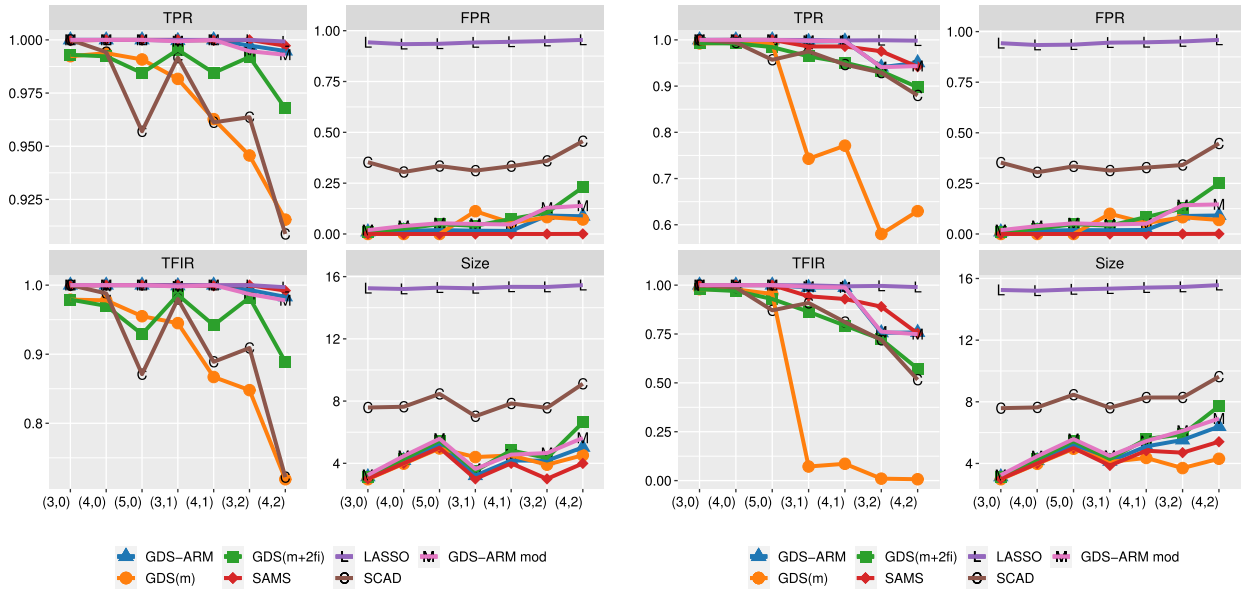

Fig. 5. Four performance metrics over 1000 iterations for $n = 24, m = 16$ when coefficients of active effects are generated from a normal distribution with standard deviation 1 and mean effect size of 5. For an active interaction, both main-effects are active (four plots on the left), or at least one main-effect is active (four plots on the right).

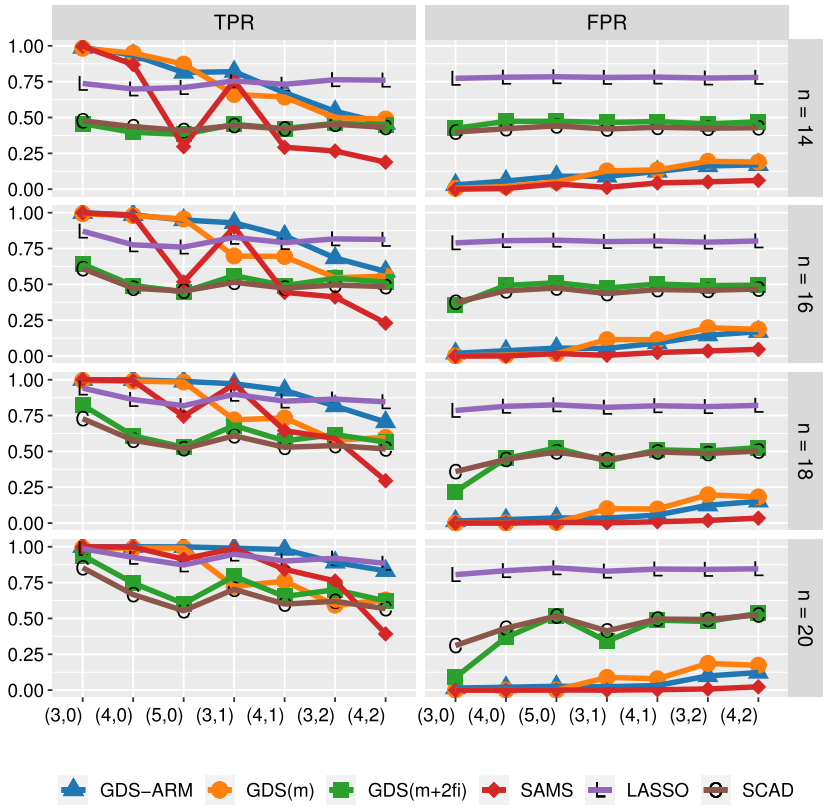

Fig. 6. Average TPR and FPR over 1000 iterations for $n = 14, 16, 18,$ and 20, $m = 24$. The sizes for the coefficients of active effects are generated from $N(5, 1)$ and at least one of the main-effects corresponding to an active interaction is also active.

which (d5) is constructed in Shi and Tang (2019), with descriptions for constructing the other designs also being provided there. These designs can also be found in the Supplementary Material. Design d5 is meant to allow for the identification of active main-effects without assuming that two-factor interactions are negligible. However, from scenarios S4-S7 in our simulations, it is clear that d5 performs poorly compared to the other designs when there are active two-factor interactions. Results for the design in Example 3 with $n = 18$ and $m = 22$ when different models for generating the data are considered are provided in the Supplementary Material. Additionally, results for the design in Example 4 with $n = 24$ and $m = 16$ using different mean effect sizes are provided in the Supplementary Material. The conclusions remain the same as described in the Examples 4 and 3, respectively. Finally, we consider designs with 24 runs and $m = 16, 20, 28$, and 32, where consistently GDS-ARM performs better than the other methods.

For almost all the cases we considered, the recommendations for tuning parameters suggested in Table 1 work. Performance of the $n = 24$ and $m = 16$ example for slightly more complicated scenarios are provided in the Supplementary Material. For situations with $n > m$, using a slightly larger value of *nint* such as $0.4\binom{m}{2}$ works a little better than the suggested choice of $0.2\binom{m}{2}$, but the choice of $0.2\binom{m}{2}$ appears to be quite robust for the values of n and m that we consider. For much larger values of n, GDS(m+2fi) is a viable method. For example, for a design with $(n, m) = (94, 20)$ (Draguljić et al., 2014), GDS(m+2fi) performs well. Using their scenarios, we studied GDS-ARM. Since the scenarios in Draguljić et al. (2014) tend to have a larger number of active interactions than in our scenarios, we need to select a larger value for *nint* for GDS-ARM to perform equivalently to GDS(m+2fi). The results in the Supplementary Material for this case use both $nint = 0.2\binom{m}{2}$ and $nint = 0.4\binom{m}{2}$, and the conclusion is that the latter choice is better. We reiterate, however, that the focus of the paper is on designs where n is much smaller and where GDS(m+2fi) is a relatively poor screening method. Based on our findings for such situations, we recommend that practitioners use the choices in the last column of Table 1 for *nrep*, *ntop*, and *pkeep*. Additionally, using $nint = 0.2\binom{m}{2}$ if $n \leq m$ and $nint = 0.4\binom{m}{2}$ otherwise works well. Note that our simulation results are based on designs with $n = 12$ to 24 and $m = 16$ to 24.

The R package `GDSARM` incorporates the GDS-ARM procedure (a) without heredity assumptions, (b) with weak heredity, and (c) with strong heredity. Though GDS-ARM is not introduced for computational gains, it is quite fast. For example, for $n = 18$, $m = 22$, it takes about (1/10)th of a second for each GDS run and hence $nrep * (1/10) = 24$ secs to run the GDS-ARM on a Desktop with an AMD Ryzen Threadripper PRO 5955WX @4.00 GHz and 64GB RAM.

6. Conclusions

For a complicated process that can be affected by many factors, one should expect that there are some active interactions. If interactions are completely ignored in factor screening, this can lead to erroneous conclusions, both through failing to select some important factors and through incorrectly selecting some factors that are not important. Due to the limited number of runs, a model with only main-effects and all two-factor interactions already becomes extremely complex due to high correlations between model columns. With a nod to the effect hierarchy principle, we must therefore assume that interactions of three or more factors are negligible. Additionally, because of effect sparsity, we focus on situations with a relatively small number of active effects, with more active main-effects than active two-factor interactions.

While GDS is a popular analysis method for screening experiments, neither GDS(m) nor GDS(m+2fi) performs well for a model with main-effects and two-factor interactions, especially when n is relatively small. We proposed a new analysis method, GDS-ARM, which identifies important factors by aggregating results from multiple GDS applications performed on models with different sets of randomly selected interactions. GDS-ARM draws its motivation in part from random forests by using models that contain only some of the available effects, and by identifying important factors after applying GDS on all of these models. Through the simulations and real case studies, we demonstrate that GDS-ARM works well across a range of scenarios, designs, and different values of n and m, and almost always outperforms other available methods. Aside from GDS(m) and GDS(m+2fi), we compared the results from GDS-ARM to those from SAMS, LASSO and SCAD. Only SAMS delivered comparable results at times, particularly when the number of runs, n, was larger and the number of factors, m, was smaller. Besides the original GDS-ARM version, we also studied a modified version in which lines 4 and 9 of Algorithm 1 were modified by not deleting interactions that violated the weak heredity principle. The modified version performed better when the generation of simulated data did not impose heredity assumptions, but had slightly higher FPR values than the original version when heredity assumptions were imposed. The choice between the original and modified version of GDS-ARM can be made based on how strongly one believes that models adhere to some form of heredity.

Our choice of tuning parameters (Table 1) focuses on the scenarios that have 0 to 2 active interactions. If it is anticipated that more interactions could be active, a different choice of *nint* could yield better results. Largely, it seems that the choice of design does not matter much. However, as seen in the Supplementary Material for $n = 18, m = 22$ and $n = 12, m = 16$, some designs can perform poorly. Further investigations are necessary to identify designs that work best for GDS-ARM. A straightforward generalization of GDS-ARM to factors with three or more levels is possible, however, more work would be needed to tune the parameters for such designs.

Funding information

JS gratefully acknowledges support through NSF grants DMS-1935729 and DMS-2304767.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2024.107940 and contains additional results and the designs used in the analysis. R codes are available in the package GDSARM available to download from CRAN.

References

Bien, J., Taylor, J., Tibshirani, R., 2013. A lasso for hierarchical interactions. Ann. Stat. 41 (3), 1111–1141.

Booth, K.H.V., Cox, D.R., 1962. Some systematic supersaturated designs. Technometrics 4 (4), 489–495.

Box, G.E.P., Meyer, R.D., 1993. Finding the active factors in fractionated screening experiments. J. Qual. Technol. 25 (2), 94–105.

Breiman, Leo, 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Candès, E., Tao, T., 2007. The Dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. 35 (6), 2313–2351.

Chipman, H., Hamada, M., Wu, C.F.J., 1997. A bayesian variable-selection approach for analyzing designed experiments with complex aliasing. Technometrics 39 (4), 372–381.

Dopico-García, M.S., Valentao, P., Guerra, L., Andrade, P.B., Seabra, R.M., 2007. Experimental design for extraction and quantification of phenolic compounds and organic acids in white “vinho verde” grapes. Anal. Chim. Acta 583 (1), 15–22.

Draguljić, 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), 1–15.

Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. 96 (456), 1348–1360.

Friedman, Jerome, Hastie, Trevor, Tibshirani, Robert, 2001. The Elements of Statistical Learning. Springer Series in Statistics, vol. 1. New York.

Hamada, M., Wu, C.F.J., 1992. Analysis of designed experiments with complex aliasing. J. Qual. Technol. 24 (3), 130–137.

Hunter, G.B., Hodi, F.S., Eagar, T.W., 1982. High cycle fatigue of weld repaired cast Ti-6AI-4V. Metall. Trans. A 13 (9), 1589–1594.

Jones, B., Lekivetz, R., Majumdar, D., Nachtsheim, C.J., Stallrich, J.W., 2020. Construction, properties, and analysis of group-orthogonal supersaturated designs. Technometrics 62 (3), 403–414.

Jones, B.A., Majumdar, D., 2014. Optimal supersaturated designs. J. Am. Stat. Assoc. 109 (508), 1592–1600.

Jones, B.A., Lin, D.K.J., Nachtsheim, C.J., 2008. Bayesian D-optimal supersaturated designs. J. Stat. Plan. Inference 138 (1), 86–92.

Li, X., Sudarsanam, N., Frey, D.D., 2006. Regularities in data from factorial experiments. Complexity 11 (5), 32–45.

Lloyd, S.P., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28 (2), 129–137.

Marley, C.J., Woods, D.C., 2010. A comparison of design and model selection methods for supersaturated experiments. Comput. Stat. Data Anal. 54 (12), 3158–3167.

McGrath, Richard N., Xu, Yuhang, Taylor, Anna, 2023. Screening main and interaction effects in a plackett-burman design. Commun. Stat., Simul. Comput., 1–21.

Phoa, F.K., Pan, Y.H., Xu, H., 2009a. Analysis of supersaturated designs via the Dantzig selector. J. Stat. Plan. Inference 139 (7), 2362–2372.

Phoa, F.K., Wong, W.K., Xu, H., 2009b. The need of considering the interactions in the analysis of screening designs. J. Chemom. 23 (10), 545–553.

Schoen, E.D., Vo-Thanh, N., Goos, P., 2017. Two-level orthogonal screening designs with 24, 28, 32, and 36 runs. J. Am. Stat. Assoc. 112 (519), 1354–1369.

Shi, C., Tang, B., 2019. Supersaturated designs robust to two-factor interactions. J. Stat. Plan. Inference 200, 119–128.

Singh, Rakhi, Stufken, John, 2023. Selection of two-level supersaturated designs for main effects models. Technometrics 65 (1), 96–104. https://doi.org/10.1080/00401706.2022.2102080.

Smucker, B.J., Drew, N.M., 2015. Approximate model spaces for model-robust experiment design. Technometrics 57 (1), 54–63.

Stallrich, Jon, Young, Kade, Weese, Maria, Smucker, Byran, Edwards, David, 2023. Optimal Supersaturated Designs for Lasso Sign Recovery.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc., Ser. B, Methodol. 58 (1), 267–288.

Vazquez, A.R., Schoen, E.D., Goos, P., 2020. A mixed integer optimization approach for model selection in screening experiments. J. Qual. Technol., 1–24.

Weese, M.L., Smucker, B.J., Edwards, D.J., 2015. Searching for powerful supersaturated designs. J. Qual. Technol. 47 (1), 66–84.

Weese, M.L., Edwards, D.J., Smucker, B.J., 2017. A criterion for constructing powerful supersaturated designs when effect directions are known. J. Qual. Technol. 49 (3), 265–277.

Weese, M.L., Stallrich, J.W., Smucker, B.J., Edwards, D.J., 2021. Strategies for supersaturated screening: group orthogonal and constrained var(s) designs. Technometrics 63 (4), 443–455.

Westfall, P.H., Young, S.S., Lin, D.K.J., 1998. Forward selection error control in the analysis of supersaturated designs. Stat. Sin. 8, 101–117.

Wolters, M.A., Bingham, D., 2011. Simulated annealing model search for subset selection in screening experiments. Technometrics 53 (3), 225–237.

Yuan, M., Joseph, V.R., Lin, Y., 2007. An efficient variable selection approach for analyzing designed experiments. Technometrics 49 (4), 430–439.

Yuan, M., Joseph, V.R., Zou, H., 2009. Structured variable selection and estimation. Ann. Appl. Stat. 3 (4), 1738–1757.