A Mixed Integer Optimization Approach for Model Selection in Screening Experiments

Alan R. Vazquez[1,2], Eric D. Schoen[1,3], and Peter Goos[1,2]

[1]Department of Biosystems, KU Leuven, Belgium
[2]Department of Engineering Management, University of Antwerp, Belgium
[3]Department RAPID, TNO, Zeist, Netherlands

May 25, 2020

Abstract

After completing the experimental runs of a screening design, the responses under study are analyzed by statistical methods to detect the active effects. To increase the chances of correctly identifying these effects, a good analysis method should provide alternative interpretations of the data, reveal the aliasing present in the design, and search only meaningful sets of effects as defined by user-specified restrictions such as effect heredity. This article presents a mixed integer optimization strategy to analyze data from screening designs that possesses all these properties. We illustrate our method by analyzing data from real and synthetic experiments, and using simulations.

Keywords: Best-subset selection, Dantzig selector, definitive screening design, sparsity, two-factor interaction.

1 Introduction

Screening experiments permit the study of many factors using a small number of runs. Successful screening requires the assumption that only a small proportion of the factors' effects on the responses of interest matter. This assumption is known as effect sparsity. Screening experiments are commonly carried out using two-level orthogonal screening designs (Mee et al., 2017; Schoen et al., 2017), three-level orthogonal screening designs (Cheng and Wu, 2001; Xu et al., 2004), mixed-level orthogonal screening designs (Wu and Hamada, 2009, ch. 7), or definitive screening designs (Jones and Nachtsheim, 2011). These designs have economical run sizes but possess complex aliasing structures that make the identification of the influential or *active* effects challenging.

In this article, we introduce an effective method to analyze data from screening designs and identify the active effects. Our method possesses various properties which are desirable for an adequate analysis of screening experiments. In the rest of this section, we introduce these properties, review the existing methods to analyze data from screening designs and state the contributions of this article.

1.1 Properties of good data analysis methods

The analysis of data from screening designs involves a model selection problem characterized by a small number of observations and a large number of effects. The goal is to find the smallest model that includes the active effects. Abraham et al. (1999) and Mee (2013) provide guidelines on the use of model selection methods to analyze data from screening designs. According to these authors' findings, a method suitable for the analysis has several specific characteristics. We formulate these characteristics as desirable properties of a good model selection method:

Property 1. A good model selection method creates a list of models that are compatible with the data.

Often, there is more than one model that explains the observed data well. Therefore, having a list of good models rather than a single overall-best model provides more information on the potentially active effects. Moreover, if the results and the analysis from the screening design used are not satisfactory, such a list of good models suggests which effects should be considered for further investigation in a follow-up experiment.

Property 2. A good model selection method reveals the aliasing present in the design.

When using a screening design to study many factors, the number of possible models is large, and, due to the limited number of runs in a screening design, many effects are aliased. In that case, some of these models will be highly aliased and therefore have a similar or even identical fit to the data. The active effects can then not be identified unambiguously. A good model selection method should reveal any aliasing present in the data and thereby recognize the limitations of the screening design used and refrain the experimenter from drawing unwarranted conclusions.

Property 3. A good model selection method allows the user to specify restrictions on the model search.

Model selection methods that possess Property 3 can restrict the search space to specific sets of meaningful models such as those obeying weak or strong effect heredity (Hamada and Wu, 1992), where it is assumed that an interaction can be active only if one or both of the corresponding main effects are also active, and that a quadratic effect can only be active if the corresponding main effect is active too. Based on a meta-analysis of a large number of two-level factorial experiments and response surface experiments, Li et al. (2006) and Ockuly et al. (2017) showed that effect heredity generally holds in practice. It therefore makes sense to incorporate heredity restrictions in model selection. Other situations in which user-specified restrictions are useful in the context of model selection include experiments involving multi-level categorical factors. In the analysis of data from such experiments, it is generally desirable to select all contrast vectors corresponding to the categorical factor simultaneously for inclusion in the model. Restrictions on the model search can also be used to set bounds on the model size or the number of factors in the model, or to avoid models that are known in advance to be misleading.

1.2 Properties of existing methods

Model selection methods available in the literature can be categorized into shrinkage and nonshrinkage methods. Shrinkage methods (Hastie et al., 2009, ch. 3) perform model selection by biasing, or shrinking, some of the effect estimates toward zero. In contrast, nonshrinkage methods perform model selection without biasing the effect estimates. Here, we briefly introduce the existing shrinkage and nonshrinkage methods to analyze data from screening designs, and conclude that none of them possesses Properties 1, 2 and 3 simultaneously. We include a detailed account of these methods in Supplementary Section A.

1.2.1 Shrinkage methods

Perhaps two of the most popular shrinkage methods to analyze data from screening designs are the LASSO (Tibshirani, 1999) and the Dantzig selector (Candes and Tao, 2007) . In recent years, multiple extensions to these two methods have been presented. For instance, Yuan and Lin (2006) and Liu et al. (2010) extended the LASSO and the Dantzig selector, respectively, to deal with categorical factors. Extensions to the LASSO to impose effect heredity can be found in Choi et al. (2010) and Bien et al. (2013). Although the primary focus of these methods and their extensions is to build good predictive models, several authors use them for analyzing data from screening designs and thus for identifying active effects (Phoa et al., 2009; Marley and Woods, 2010; Draguljić et al., 2014; Weese et al., 2015; Errore et al., 2017).

Shrinkage methods include a tuning parameter that controls the degree of shrinkage in the estimates and the complexity of the model. Models obtained for different values of the tuning parameter provide alternative interpretations of the data and imply that shrinkage methods possess Property 1. A drawback of shrinkage methods for analyzing data from screening designs is that they do not have the potential to reveal strong aliasing patterns due to the design. For instance, for a fixed value of the tuning parameter, these methods cannot generate alternative models supported by the data. Another limitation of these methods is that they do not allow any user-specified search constraint other than heredity constraints (which are available for the LASSO only) to be incorporated in the model search. For all these reasons, shrinkage methods do not possess Properties 2 and 3.

1.2.2 Nonshrinkage methods

Two popular nonshrinkage methods for analyzing data from screening designs are forward selection (Westfall et al., 1998) and simulated annealing model search (SAMS; Wolters and Bingham, 2011). Forward selection builds a model sequentially by adding the most significant effect at each step. The process finishes when no more significant effects can be added to the model. The main advantages of this approach are that it is computationally fast and produces the ordinary least squares estimates for the selected effects. Three of the most popular implementations of forward selection are available in SAS v9.4, JMP v13 and the package ‘leaps’ v2.9 in R. However, these implementations are either not suitable to create a list of good models, to assess the aliasing of effects or to incorporate restrictions in the model search, other than effect heredity and restrictions to deal with categorical factors. Therefore, forward selection, as implemented in these packages, does not possess Properties 1, 2 and 3, simultaneously.

SAMS utilizes a heuristic algorithm, called the simulated annealing algorithm, to find a large list of models (usually 10,000) that fit the data well or reasonably well and which are explored graphically. The algorithm constructs models that obey effect heredity and have a fixed number of terms, which is up to four units larger than the largest number of effects assumed to be active. SAMS uses graphical aids, called raster plots, to assess the aliasing caused by the experimental design and to detect the active effects. Due to its simulated annealing algorithm and its graphical aids, SAMS possesses Properties 1 and 2. However, the current implementation of SAMS cannot handle data from experiments involving categorical factors with three levels or more, and it does not allow model search restrictions other than heredity for two-factor interactions or quadratic effects to be imposed. Modifying the simulated annealing algorithm to overcome these shortcomings is not trivial, so that, at present, SAMS does not possess Property 3.

1.3 Contribution and organization

In this article, we present a model selection method to analyze data from screening designs that possesses Properties 1, 2 and 3, introduced in Section 1.1. Our method builds upon the recent work of Bertsimas et al. (2016) and Bertsimas and King (2016) on best-subset selection (Miller, 2002) and uses modern mixed integer optimization methods from the field of operations research to find high-quality models of any size. We introduce our method, called MIO because of its use of Mixed Integer Optimization, and discuss its strengths in Section 2. In Section 3, we illustrate the effectiveness of MIO by analyzing data from synthetic and real screening experiments. In Section 4, we present a simulation study to compare the performances of MIO and the benchmark methods discussed in Section 1.2, when it comes to correctly identifying active effects. We conclude the article and mention avenues for future research in Section 5. The supplementary materials of this article include additional sections where we provide a comprehensive comparison between the benchmark methods and our method, as well as a Python implementation of MIO.

Throughout the article, we consider a linear regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{y}$ is an $n \times 1$ vector of responses and n is the number of observations, $\boldsymbol{\beta}$ is a $p \times 1$ vector of p unknown parameters, $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of independent and normally distributed random errors with zero mean and variance σ^2, and $\mathbf{X}$ is an $n \times p$ model matrix. Note that $\mathbf{X}$ can include contrast vectors associated with the main effects (MEs), two-factor interactions (2FIs) and quadratic effects (QEs) of quantitative factors as well as contrast vectors associated to the effects of multi-level categorical factors. We assume that the columns of $\mathbf{X}$ have been standardized to have zero means and to have the same length, and that $\mathbf{y}$ is centered around zero to exclude the intercept from the model. We denote the element in the ith row and uth column of $\mathbf{X}$ by x_{iu}. Similarly, we denote the ith element of $\mathbf{y}$ by y_i.

2 MIO in full

For any given model size up to a user-specified maximum, $k_{\max}$, MIO lists the best models in terms of the residual sum of squares (RSS). For a given size, the list contains the top M models that satisfy any user-specified model search restrictions. Due to the fact that it produces a list of models, MIO possesses Properties 1 and 2. Because of its ability to include model search restrictions, MIO also possesses Property 3. A key feature of MIO is that it also visualizes the list of models using raster plots, which allows for detecting patterns in the selected effects. The most important factor effects are those that appear consistently in the best models. These can then be declared active.

We first describe the core mixed integer optimization procedure used by MIO to find the best-fitting models. Next, we show how to incorporate user-specified restrictions in the model search and then embed the MIO approach in a sequential algorithm to list the best M feasible models for any given model size. Finally, we introduce the raster plots and discuss the implementation of our MIO approach.

2.1 MIO's optimization problem

2.1.1 Basic idea

For a given model size k, MIO searches for the model that minimizes the RSS value. In other words, it seeks the model that has the best least-squares fit to the data and thus performs a best-subset selection (Miller, 2002). Essentially, best-subset selection solves the following problem:

$$\min_{\hat{\boldsymbol{\beta}} \in \mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \sum_{u=1}^{p} x_{iu} \hat{\beta}_u \right)^2 \text{ subject to } \sum_{u=1}^{p} I(\hat{\beta}_u \neq 0) \leq k, \tag{1}$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p)^T$ is the vector of parameter estimates and $I(c)$ is an indicator function that takes the value 1 if condition c is satisfied and 0 otherwise. The nonzero $\hat{\beta}_u$ values correspond to the ordinary least squares (OLS) estimates of the parameters corresponding to the selected model terms. The constraint in the optimization problem ensures that at most k terms are selected for inclusion in the regression model. More specifically, it ensures that at most k model parameters are nonzero. The parameter k thus has a simple interpretation, and the parameter estimates produced are plain OLS estimates. So, MIO does not use any shrinkage.

In technical terms, the best-subset selection problem in (1) is an NP-hard problem (Natarajan, 1995), which means that it cannot be solved in polynomial time. Popular state-of-the-art algorithms to tackle this problem are available in SAS v9.4, JMP v13 and the package 'leaps' v2.9 in R. The most computationally efficient of these algorithms are those implemented in SAS v9.4 and the package 'leaps' v2.9, since they avoid a complete enumeration of all possible models by using the leaps and bounds algorithm of Furnival and Wilson (1974), which combines computationally efficient matrix operators with a branch and bound procedure. Despite these attractive features, the leaps and bounds algorithm does not allow the problem to be solved when it involves more than 60 potential effects and a moderate model size. Therefore, in spite of its intuitive appeal, best-subset selection is currently considered infeasible for screening experiments involving many factors. However, due to its use of the work Bertsimas et al. (2016) as well as modern optimization methods in state-of-the-art solvers such as Gurobi, CPLEX or SCIP, our proposed MIO approach makes best-subset selection feasible for a broader range of screening designs than it is currently possible with the available algorithms.

2.1.2 Problem formulation

Mixed integer optimization is an optimization method to determine the values of a set of discrete and continuous decision variables so as to maximize or minimize a particular linear or quadratic objective function, while satisfying a set of linear or quadratic constraints (Bertsimas and Weismantel, 2005). Solvers such as Gurobi, CPLEX or SCIP can be used to tackle mixed integer optimization problems. The solvers provide both feasible solutions and bounds for the objective function's optimal value. As the solvers progress toward the optimal solution, the bounds improve and provide an increasingly better guarantee of optimality, which is especially useful if the solver is stopped before it converges to the global optimum. In contrast, heuristic algorithms do not provide such a certificate of optimality.

Bertsimas et al. (2016) formulated the best-subset selection problem in (1) as a mixed integer optimization problem to solve large instances with associated certificates of optimality, using a computational time that is acceptable for practical applications. Their approach is a dramatic improvement over the leaps and bounds algorithm of Furnival and Wilson (1974) due to recent developments in computer hardware and to both theoretical and practical advances in mixed integer optimization such as cutting plane theory, disjunctive programming for branching rules, and improved heuristic and linear optimization methods (Bixby, 2012). Bertsimas et al. (2016) present two MIO formulations. One is intended for problems in which the number of parameters p is smaller than the number of runs n, while the other is for problems in which the number of runs n is smaller than the number of parameters p. We adopt the latter version because of the fact that $n < p$ is a basic characteristic of screening experiments, especially when considering 2FIs in addition to MEs. Therefore, our MIO problem is the following:

$$\min_{\hat{\boldsymbol{\beta}},\hat{\mathbf{y}},\mathbf{z}} \hat{\mathbf{y}}^T\hat{\mathbf{y}} - 2(\mathbf{X}^T\mathbf{y})^T\hat{\boldsymbol{\beta}} + \mathbf{y}^T\mathbf{y} \tag{2}$$

$$\text{subject to}$$

$$\text{Model constraints:}$$

$$(1 - z_u) = 0 \text{ or } \hat{\beta}_u = 0, \quad u = 1, \dots, p, \tag{3}$$

$$\sum_{u=1}^{p} z_u \leq k, \tag{4}$$

$$\text{Technical constraints:}$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}, \tag{5}$$

$$z_u \in \{0, 1\}, \quad u = 1, \dots, p, \tag{6}$$

$$\text{Boosting constraints (optional):}$$

$$-B \leq \hat{\beta}_u \leq B, \quad u = 1, \dots, p, \tag{7}$$

$$\sum_{u=1}^{p} |\hat{\beta}_u| \leq B^L, \tag{8}$$

$$-E \leq \hat{y}_i \leq E, \quad i = 1, \dots, n, \tag{9}$$

$$\sum_{i=1}^{n} |\hat{y}_i| \leq E^L. \tag{10}$$

In this problem formulation, $\hat{\beta}_u$ represents the uth element of $\hat{\boldsymbol{\beta}}$, z_u is a binary variable associated with it, $\hat{y}_i$ is the ith element of the $n \times 1$ vector $\hat{\mathbf{y}}$, $\mathbf{z} = (z_1, z_2, \dots, z_p)^T$, and B, B^L, E and E^L are auxiliary constants larger than zero. This formulation involves $p + n$ continuous decision variables contained within $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{y}}$, p binary decision variables z_u and $3p + 2n + 3$ constraints. The binary variable z_u is 0 when the corresponding parameter $\hat{\beta}_u$ is set to 0, because the uth column of the model matrix $\mathbf{X}$ does not belong to the best subset. It takes the value 1 when the term corresponding to the uth column of $\mathbf{X}$ does belong to the best subset, with an effect estimate equal to $\hat{\beta}_u$.

The objective function (2) expresses the RSS value in the objective function in problem (1) in terms of the vector of effect estimates $\hat{\boldsymbol{\beta}}$ and the vector of predicted responses $\hat{\mathbf{y}}$. In the new objective function, the vector $\mathbf{X}\hat{\boldsymbol{\beta}}$ is replaced by $\hat{\mathbf{y}}$, so that the number of quadratic terms in the objective function is n rather than p. This feature improves the computing time required by the solver when p is much larger than n, which is typical for screening experiments.

The MIO problem formulation involves three kinds of constraints: model constraints, technical constraints and boosting constraints. There are two types of model constraints. The first type of constraint in (3) states that $\hat{\beta}_u$ or $1 - z_u$ is zero. In other words, it implies that either the uth model term is selected in the best subset and can have a nonzero parameter estimate, or that the uth term is not selected and has a zero parameter estimate. Solvers such as Gurobi, CPLEX and SCIP can handle the type of constraint in (3) by transforming it into a specially ordered set of type 1 (SOS_1; Beale and Forrest, 1976). An SOS_1 is a set of decision variables at most one of which can be different from zero. Using these sets generally speeds up the branch and bound algorithms implemented in the solvers; see Bertsimas et al. (2016). The second type of model constraint in (4) ensures that at most k model terms are selected and can have a nonzero parameter estimate.

The technical constraint in (5) takes care of the substitution of vector $\mathbf{X}\hat{\boldsymbol{\beta}}$ with $\hat{\mathbf{y}}$ in the objective function. The constraints in (6) ensure that the variables z_u are binary.

The boosting constraints in (7)–(10) are optional: we do not need these constraints for the MIO approach to provide optimal solutions. However, the boosting constraints avoid the need to explore all possible real values for the elements of $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{y}}$, which significantly improves the computing time required by the solver to certify optimality. The constants B and E bound the absolute $\hat{\beta}_u$ and $\hat{y}_i$ values, respectively, while B^L and E^L bound the sums of the absolute $\hat{\beta}_u$ and $\hat{y}_i$ values. In the Appendix, we show how to specify accurate values for B, E, B^L and E^L.

After solving the MIO problem to optimality, the output is twofold. First, the nonzero z_u values indicate the best subset, i.e., the set of k model terms that minimize the RSS value. Second, the corresponding $\hat{\beta}_u$ values are the OLS estimates of the parameters corresponding to the selected model terms. Note that, if the data set allows the estimation of a model with k terms, an optimal solution to the MIO problem in (2)–(10) will generally involve the maximum number of effects, k, specified by constraint (4). This is because adding a term to a model results in a RSS value that is at least as good as the previous one.

2.2 Search restrictions

Bertsimas and King (2016) explained that the MIO problem can include linear constraints to incorporate subject-matter expertise in the model selection. These constraints have the form $\mathbf{a}^T\mathbf{z} \leq b$, where $\mathbf{a}$ is a $p \times 1$ vector, $\mathbf{z}$ is the $p \times 1$ vector of the binary decision variables z_u, and b is a constant. In this section, we show that, due to the possibility of incorporating such constraints, MIO possesses Property 3 of a good model selection method. More specifically, we show that the constraints allow us to impose weak and strong effect heredity, to limit the number of factors in a model, to properly deal with multi-level categorical factors and to exclude specific models from the model search.

2.2.1 Effect heredity

Restricting the model search to the class of models obeying effect heredity is a well-established practice among practitioners searching for active second-order effects, i.e., 2FIs and QEs. The meta-analyses of Li et al. (2006) and Ockuly et al. (2017) of large numbers of published data sets from two-level factorial and response surface experiments, respectively, provide empirical support for the use of effect heredity in model selection. More specifically, both studies revealed that second-order effects are much more likely to be active when their corresponding MEs are active. This implies that, when searching for good models, it is generally a waste of effort to study models including certain 2FIs and QEs, but not the corresponding MEs.

Two types of effect heredity are commonly used: strong and weak heredity. Under strong heredity, a 2FI can be included in the model only if both of the corresponding MEs are considered active and therefore included in the model as well. Under weak heredity, a 2FI can be included in the model when at least one of the corresponding MEs is considered active and included in the model too. Under both strong and weak heredity, the QE of a factor can be considered for inclusion in the model provided its ME has already been included because it is considered active. So, for QEs, weak and strong heredity are equivalent.

To see how heredity can be imposed in the MIO approach, denote the number of factors under investigation by m, the binary variable associated with the 2FI between the factors u and v by z_{uv}, the binary variable associated with the quadratic effect of factor u by z_{uu}, and the binary variables associated with the MEs of the factors u and v by z_u and z_v, respectively. To impose strong heredity for the 2FIs, we should add the constraints

$$z_{uv} \leq z_u \text{ and } z_{uv} \leq z_v, \quad u = 1, \ldots, m-1;\ v = u+1, \ldots, m, \tag{11}$$

to the MIO formulation in (2)–(10). To impose weak heredity for the 2FIs, we should add the constraints

$$z_{uv} \leq z_u + z_v, \quad u = 1, \ldots, m-1;\ v = u+1, \ldots, m. \tag{12}$$

To impose (weak or strong) heredity for the QEs, we need to add the constraints

$$z_{uu} \leq z_u, \quad u = 1, \ldots, m. \tag{13}$$

The meta-analysis of data from response surface experiments by Ockuly et al. (2017) indicated that 2FIs are also more likely to be active when the QEs of the factors involved are active. This motivated these authors to introduce the concepts of strong and weak quadratic/interaction heredity. Under strong quadratic/interaction heredity, a 2FI can be included in the model only if the QEs of both factors involved are considered active and therefore included in the model. Under weak quadratic/interaction heredity, a 2FI can be included in the model when the QE of at least one of the factors is considered active and therefore included in the model. By adding the constraints

$$z_{uv} \leq z_{uu} \text{ and } z_{uv} \leq z_{vv}, \quad u = 1, \ldots, m-1;\ v = u+1, \ldots, m, \tag{14}$$

to the MIO formulation, we can enforce strong quadratic/interaction heredity in the search for good models, while, by adding the constraints

$$z_{uv} \leq z_{uu} + z_{vv}, \quad u = 1, \ldots, m-1;\ v = u+1, \ldots, m, \tag{15}$$

we can enforce weak quadratic/interaction heredity.

2.2.2 Factor sparsity

The results of screening experiments can be analyzed effectively only if the assumption of effect sparsity holds. According to this assumption, just a small proportion of the factors' effects on the responses of interest matter. Sometimes, it is assumed instead, or in addition, that only a limited number of factors drive the responses. This assumption is known as factor sparsity (Box and Meyer, 1986).

We can embed factor sparsity in the MIO approach by imposing an upper bound on the number of factors that can be included in the selected model. This requires additional constraints, involving a new kind of binary decision variable w_u, which takes the value 1 if the uth factor is included in the model and the value 0 if it is not. For instance, for a continuous factor u, w_u should take the value 1 as soon as the ME, one of the 2FIs or the QE of that factor enter the model. To this end, we can add the following five types of constraints to a MIO formulation involving only continuous factors:

$$\begin{aligned} z_u &\leq w_u, \quad u = 1, \dots, m, & (16) \\ z_{uv} \leq w_u \text{ and } z_{uv} &\leq w_v, \quad u = 1, \dots, m-1;\ v = u+1, \dots, m, & (17) \\ z_{uu} &\leq w_u, \quad u = 1, \dots, m, & (18) \\ z_u + \sum_{v=1}^{u-1} z_{vu} + z_{uu} + \sum_{v=u+1}^{m} z_{uv} &\geq w_u, \quad u = 1, \dots, m, & (19) \end{aligned}$$

and

$$w_u \in \{0, 1\}, \quad u = 1, \dots, m. \tag{20}$$

The key constraint imposing factor sparsity is given by

$$\sum_{u=1}^{m} w_u \leq f, \tag{21}$$

where f is the maximum number of factors that is allowed to enter the model.

It is possible to simultaneously impose effect sparsity and factor sparsity in the MIO approach, by incorporating the constraint in Equation (4) (which imposes effect sparsity) as well as the constraints in Equations (16)–(21) (which impose factor sparsity) simultaneously.

2.2.3 Categorical factors

The levels of an l-level categorical factor are coded using a set of $l-1$ contrast vectors. So, adding the ME of an l-level categorical factor to a model implies the addition of $l-1$ terms and the simultaneous estimation of $l-1$ additional parameters. To ensure that the MIO model selection procedure either enters all $l-1$ terms in the model or none of them, we need to add the following constraint to the MIO problem:

$$z_{j_1} = \cdots = z_{j_{l-1}}, \tag{22}$$

where $j_1, \dots, j_{l-1}$ identify the columns of $\mathbf{X}$ containing the $l-1$ contrast vectors associated with the l-level categorical factor, and j_r denotes the rth contrast vector of the factor. We refer to (22) as a grouping constraint. The grouping constraint ensures that the estimates of the $l-1$ parameters corresponding to the categorical factor's ME are either all zero at the same time or not.

Grouping constraints can be used together with heredity constraints to impose strong or weak effect heredity for categorical factors. For instance, let G_a denote the set of columns of $\mathbf{X}$ containing the contrast vectors associated with an l_1-level categorical factor a, G_b denote the set of columns of $\mathbf{X}$ containing the contrast vectors associated with an l_2-level categorical factor b, and $G_{a\times b}$ denote the set of columns of $\mathbf{X}$ containing the $(l_1-1)(l_2-1)$ contrast vectors associated with the 2FI involving a and b. Strong effect heredity can then be imposed by adding two constraints of the type

$$z_q \leq z_i \text{ and } z_q \leq z_j, \tag{23}$$

where $q \in G_{a\times b}$, $i \in G_a$ and $j \in G_b$, to the MIO problem formulation, together with the grouping constraints for G_a, G_b and $G_{a\times b}$. Which decision variables z_q, z_i and z_j are used in this constraint does not impact the final solution, provided they correspond to elements of $G_{a\times b}$, G_a and G_b, respectively. The constraints in (23) imply that the $(l_1-1)(l_2-1)$ interaction terms can only be entered into the model when the MEs of the two factors involved are considered active and all $(l_1-1)+(l_2-1)$ ME terms are therefore included in the model. To impose weak heredity rather than strong heredity, we need add one constraint of the type $z_q \leq z_i + z_j$.

We can also develop expressions similar to (16)–(19) which, together with grouping constraints, impose factor sparsity for categorical factors. For a categorical factor a, w_a should take the value 1 as soon as the l_1-1 terms corresponding to its ME or the $(l_1-1)(l_2-1)$ terms corresponding to a 2FI with another categorical factor b having l_2 levels, enter the model. This can be achieved by including the grouping constraints for G_a, G_b and $G_{a\times b}$ in the MIO formulation, together with constraint (21) and each of the following three types of constraints:

$$\begin{array}{rcll} z_i & \leq & w_a, & i \in G_a, \\ z_q \leq w_a \text{ and } z_q & \leq & w_b, & q \in G_{a\times b},\ \forall b \neq a, \\ z_i + \displaystyle\sum_{\forall b;\ q \in G_{a\times b}} z_q & \geq & w_a, & i \in G_a, \end{array}$$

and

$$w_a \in \{0,1\}.$$

2.2.4 Subset constraints

In some cases, we may want to exclude certain models from the space of models MIO explores. For instance, we may want to avoid models or combinations of effects that are known in advance to be misleading. Also, after having identified the overall best model of a given size, we may want to identify the second best model of that size. This can be done by solving the MIO problem with an additional constraint that states that the overall best model can no longer be selected.

To achieve this, we have to define a subset of terms that cannot enter the model simultaneously. If we denote that subset by S and its cardinality by $|S|$, then the constraint

$$\sum_{u \in S} z_u \leq |S| - 1 \tag{24}$$

ensures that not all terms in S can be included in the model simultaneously. We refer to (24) as a subset constraint. In Section 2.3, subset constraints play a key role in our algorithmic approach to create lists of best-fitting models.

In certain cases, it may be useful to consider only models that contain a specific set of terms. For example, when analyzing data from a blocked experiment, it is recommended to include the main effect(s) of the blocking factor(s) in all models under investigation. In such cases, we need to define another set, say S', of terms that must be included in the model, and add the following constraint to the MIO formulation:

$$\sum_{u \in S'} z_u = |S'|.$$

2.3 A sequential MIO algorithm to list the best models

When searching for the active effects in screening experiments, we often desire a list of the best-fitting models instead of the overall best-fitting model. This is because there may be several models that are equally good or almost equally good in terms of the RSS. The MIO approach is ideal for creating a list of the M best-fitting models of given sizes, without enumerating and evaluating all possible models. To this end, the MIO approach uses a sequential algorithm which is based on adding subset constraints sequentially to the MIO formulation, so that previously selected models are removed from the model search. The sequential algorithm ensures that the MIO approach possesses Properties 1 and 2 of a good model selection method, because the lists of the M best models for the various k values provide alternative interpretations of the data and highlight the aliasing of the effects. More specifically, if two effects frequently alternate between the models in the list, this indicates that these effects are strongly or completely aliased.

2.3.1 The algorithm

The outline of our sequential MIO algorithm is shown in Algorithm 1. Besides the model matrix and the response vector, the input to the algorithm consists of a MIO formulation (which may include user-specified search restrictions, for instance to impose some form of heredity), the maximum model size $k_{\max}$, and the number of alternative models, M, to be generated for each model size. The algorithm begins by initializing the list of models L and the value of k, which indicates the current model size (see lines 1 and 2 in Algorithm 1). Next, the algorithm initializes a set of subset constraints denoted by C and starts to find the M best models with one term. To this end, the algorithm first solves the original MIO problem for $k = 1$, and adds the best-fitting model with one term to the list L (see lines 7 and 8 in Algorithm 1). The algorithm then adds a subset constraint of the type (24) involving the best-fitting model with one term to the set of constraints C (see line 9). Solving the MIO problem with the additional constraint in C excludes the best-fitting model with one term from the search space and leads to the second best-fitting model with the same number of terms (see line 12), which is also added to the list L and for which a new subset constraint is added to the set of constraints C. Solving the MIO problem with the two extra constraints in C produces the third best-fitting model with one term. This procedure continues until the M best-fitting models with one term have been identified (see lines 10–14 in Algorithm 1). The list with M models for $k = 1$ is thus obtained by solving M different MIO formulations, each with one extra subset constraint.

As soon as the exploration of models with one term is finished, the algorithm shifts its attention to models with two terms and finds the M best-fitting models with two terms. To this end, the algorithm re-initializes the set of subset constraints C. Next, it solves M different MIO formulations with an increasing number of subset constraints in C, this time for $k = 2$. The whole procedure is repeated until the M best models with $k_{\max}$ model terms have been found. The output of the algorithm is the list L containing M models for each of $k_{\max}$ different model sizes.

In practice, for most data sets and MIO formulations including user-specified search restrictions, the number of terms in the optimal solution of the MIO problem will generally have exactly k terms. One exception may be when dealing with data sets involving categorical factors, as an optimal solution to the MIO problem with the grouping constraints in

Algorithm 1: Sequential algorithm to generate lists of optimal models.

```
Input: $\mathbf{X}$, $\mathbf{y}$, MIO formulation, $k_{\max}$ and $M$
Set $L \leftarrow \varnothing$
Set $k \leftarrow 0$
while $k < k_{\max}$ do
    $k \leftarrow k + 1$
    Set $i \leftarrow 1$
    Set $C \leftarrow \varnothing$
    $S_{k,i} \leftarrow$ optimal subset obtained by solving the MIO formulation with current $k$
      value for the data set given by $\mathbf{X}$ and $\mathbf{y}$.
    $L \leftarrow L \cup \{S_{k,i}\}$
    Add the constraint $\sum_{u \in S_{k,i}} z_u \leq k - 1$ to the set of constraints $C$.
    while $i < M$ do
        $i \leftarrow i + 1$
        $S_{k,i} \leftarrow$ optimal subset obtained by solving the MIO formulation with
          current $k$ value and including all constraints in $C$ for the data set given by
          $\mathbf{X}$ and $\mathbf{y}$.
        $L \leftarrow L \cup \{S_{k,i}\}$
        Add the constraint $\sum_{u \in S_{k,i}} z_u \leq k - 1$ to $C$.
Output: List $L$ of $M$ best models for each $k$.
```

(22) may involve fewer than k terms. For example, consider a model selection problem involving the MEs of two three-level categorical factors and assume that grouping constraints have been added to the MIO formulation, to ensure that the two terms corresponding to each ME are either included in the model together or not. In that case, it is only possible to construct models of sizes two and four. Solving the MIO formulation with a k value of three will then produce a model that involves only two terms and which will be the same as the optimal two-term model. To avoid repeated models in the list of best models, the sequential MIO algorithm only reports the unique best models with k terms, which means that, in some specific cases, the list of best models may include less than $Mk_{\max}$ models.

Two strengths of our sequential algorithm are that it avoids the need for a complete enumeration of all possible models and guarantees that the M best models are found for each value of k. Compared to existing algorithms for best-subset selection, our algorithm has the advantage that the list of models satisfies any user-specified search restrictions.

2.3.2 Guidelines to specify the maximum model size

Ideally, the value of $k_{\max}$ is obtained from subject-matter experts based on their interpretation of effect sparsity. However, in the absence of subject-matter expertise, several guidelines are available in the literature. Based on a simulation study involving two-level screening designs, Marley and Woods (2010) argue that the number of observations should be at least three times the number of possibly active effects. We can therefore set $k_{\max} = \lfloor n/3 \rfloor$ when analyzing data from an n-run screening design, as we should not be expect to successfully identify more than $\lfloor n/3 \rfloor$ active effects with only n observations. Wolters and Bingham (2011) suggest adding 2, 3 or 4 units to that $k_{\max}$ value, so that overfitted models are also explored. Miller and Sitter (2001) demonstrated that a two-level screening design that permits the estimation of all models including the MEs and up to h 2FIs, can correctly identify up to $\lfloor h/2 \rfloor$ active 2FIs. So, for these designs, we can set $k_{\max}$ to $m + \lfloor h/2 \rfloor$, where m is the number of MEs.

2.4 Raster plots

After obtaining the list L of the M best-fitting models for the different model sizes k, we recommend visualizing the models by means of a raster plot. Raster plots were introduced in the literature on the analysis of screening experiments by Wolters and Bingham (2011), and they complement our sequential MIO algorithm very well. For this reason, we view raster plots as a key component of our MIO approach. A raster plot shows the effects considered on the horizontal axis and the models from the list L on the vertical axis. The models on the vertical axis are ranked according to the RSS value. The best-fitting models are located at the bottom of the raster plot. Additionally, the largest effect estimates in absolute value are visualized by the darkest cells in the plot.

2.5 MIO implementation

We implemented our MIO approach in Gurobi v.7.5 and Python. Our implementation includes the MIO formulation in Equations (2)–(10), the sequential MIO algorithm and raster plots. Moreover, by default, it uses the boosting constraints of the MIO problem and specifies the value of their constants as described in the Appendix. The implementation of our MIO approach is included in the supplementary materials accompanying this article.

To carry out the analyses of the data sets in this article, we used a standard CPU with an Intel(R) Core(TM) i7 processor with 3.4Ghz and 16 GB of RAM. So, the computing times we report for analyzing the data sets using our MIO approach are based on this hardware specification.

3 Proofs of concept

In this section, we demonstrate the power of the MIO approach by analyzing data from two synthetic experiments and two real-life experiments. The analysis of the data from the first synthetic experiment, involving a three-level design, illustrates how Properties 1, 2 and 3 of the MIO approach help to identify the active effects. The analysis of data from the second synthetic experiment, which involves the same three-level design, shows that the MIO approach has the potential to identify active 2FIs which violate effect heredity, even when heredity constraints are included in the MIO formulation. The next demonstration, involving a real-life experiment, shows the computational power of the MIO approach to analyze data from a two-level design with a large number of runs and factors. Finally, using a second real-life experiment, involving a design with seven two-level factors and two four-level categorical factors, we demonstrate that the MIO approach can deal with multi-level categorical factors. The variety in the examples considered in this section demonstrates the flexibility of the MIO approach.

Example 1. We simulated data using a definitive screening design (DSD; Jones and Nachtsheim, 2011) involving 21 runs and ten quantitative factors, which we label using the letters A–J. Table 1 shows the treatment combinations of the design, along with the simulated responses in column Y_1. The DSD has orthogonal columns for the MEs, and these columns are orthogonal to all columns of $\mathbf{X}$ that correspond to second-order effects. The maximum absolute correlation between two second-order effect columns of $\mathbf{X}$ is 3/4, implying that some of the second-order effects are strongly aliased. We simulated the responses in column Y_1 of Table 1 using the following model:

$$Y_{1i} = 2\text{A} + 2\text{C} + 2\text{BC} + \text{CD} + 4\text{C}^2 + 4\text{D}^2 + \epsilon_i, \tag{25}$$

where $\epsilon_i \sim N(0, 0.5^2)$. This model has two large QEs, namely C^2 and D^2, the former of which obeys effect heredity. Both 2FIs in the model obey weak effect heredity, and the interaction involving C and D obeys strong quadratic/interaction heredity. The detection of the active effects in model (25) is challenging because of the aliasing among the second-order effects in the DSD and the large number of effects typically considered when analyzing data from a 10-factor DSD: 10 MEs, 45 2FIs and 10 QEs.

We applied the sequential algorithm from Section 2.3 to the data in column Y_1 of Table 1 with $k_{\max} = \lfloor n/3 \rfloor + 3 = 10$, following the advice of Wolters and Bingham (2011), and generated the $M = 10$ best models for each value of $k \in \{1, \ldots, 10\}$. Initially, we utilized the original MIO formulation in (2)–(10), without adding any extra constraints. So, initially, we did not impose heredity. The sequential algorithm required 92 min of computing time to generate the complete list of the best 100 models. We compared the performance of our algorithm to the package 'leaps' v2.9 by generating the same list of models using the best-subset selection algorithm implemented in the *regsubsets* function of the package. Although the *regsubsets* function finished the search within five minutes, the resulting list contained sub-optimal models with worse RSS values than the models found by our sequential algorithm. This unexpected finding implies that our sequential algorithm is more reliable for best-subset selection than the package 'leaps' v2.9.

Figure 1a shows the raster plot for the effect estimates in each of the 100 models found by the sequential MIO algorithm. The raster plot shows that the estimates of the MEs of A and C are consistently large in the best models. Therefore, we can declare these MEs active. The successful detection of these effects is, in part, due to the good aliasing properties for the MEs in the DSD. Other effects that are frequently included in the best models are the truly active second-order effects (BC, CD, C^2 and D^2) in addition to the

2FIs AD, BG and BH. All other effects either have small estimates or appear in very few of the good models. Interestingly, a close inspection of the raster plot reveals that the best models either include the second-order effects BC, CD and D^2, or the effects AD, BG and BH. This is clearer from Figure 1b, which displays a reduced raster plot including only the estimates of these six second-order effects, in addition to the estimates of the MEs of A and C. Figure 1b shows that the models include effects from either the set {BC, CD, D^2} or {AD, BG, BH}, or do not include any of these second-order effects. Examining these sets reveals that their effects are highly aliased when using the DSD.

To prevent non-hereditary models from entering the list of good models, we also applied the MIO approach with the heredity constraints for the 2FIs and the QEs in (12) and (13), respectively, and the weak quadratic/interaction constraints in (15) for the 2FIs. The computing time required by the sequential algorithm to generate the new list was considerably smaller than when generating the list of non-hereditary models. More specifically, the algorithm only required seven minutes to generate the list of the best models satisfying the heredity constraints. The resulting raster plot in Figure 2 clearly identifies the truly active effects. The plot greatly benefits from the search restrictions as it more consistently identifies the true model. Note that, due to the heredity constraints, the models including D^2 must also include the (inactive) ME of D. So, imposing heredity forces the MIO approach to incorporate an inactive effect in the model. We do not consider this problematic because the light color corresponding to the ME estimates for factor D in the raster plot suggests that the ME is close to zero and should therefore not be declared active.

This example showed the potential of the MIO approach to generate a list of models compatible with the data (Property 1), elucidate the aliasing in the effects (Property 2) and use model search restrictions (Property 3) when identifying active effects.

Example 2. We also studied the performance of the MIO approach to correctly identify models which violate effect heredity, when, mistakenly, effect heredity constraints are included in the MIO formulation. To this end, we simulated another set of responses for the 10-factor 21-run DSD in Table 1. Inspired by Wolters and Bingham (2011), we simulated the vector of responses labeled Y_2 in Table 1 using the following model:

$$Y_{2i} = \mathrm{A} + 2\mathrm{BC} - \mathrm{CD} + \epsilon_i, \tag{26}$$

where $\epsilon_i \sim N(0, 0.5^2)$. This model includes the ME of the factor A and two 2FIs, which share factor C and do not follow effect heredity. The largest (in absolute value) of those 2FIs involves the factors B and C and has a positive coefficient. The other interaction effect, involving the factors C and D, is only half as large (in absolute value) and has a negative coefficient.

We analyzed the data using the MIO approach with the heredity constraints for the 2FIs and the QEs in (12) and (13), respectively. We again set the value of $k_{\max}$ to 10 as in the previous example and generated the $M = 10$ best models for each $k \in \{1, \ldots, 10\}$. The computing time required by the sequential MIO algorithm to generate the list of the 100 best models was 61 min. Figure 3 shows the raster plot for the effect estimates in each of the models in this list. The raster plot clearly shows that the ME of A and the 2FIs involving B and C, and C and D, consistently appear in the best models. The raster plot also shows that the estimates of these effects are the largest ones in absolute value, since they correspond to the darkest cells in the plot. Other effects that consistently appear in the best models are the MEs of B and D, which are inactive. These effects are included to satisfy the requirement that the models must follow weak effect heredity, whenever they include the interaction BC or CD. However, the light colored cells corresponding to these effects' estimates in the plot suggest that they are close to zero and thus should not be declared active.

Another effect that consistently appears in the best models is the 2FI involving B and D. The estimate for this 2FI is larger (in absolute value) than the estimates for the MEs of B and D, since the cells corresponding to that 2FI in the raster plot are darker than those corresponding to the MEs. So, this inactive 2FI might incorrectly be considered as active by the MIO approach. We believe this is a small price to pay for the fact that the MIO approach with heredity constraints was able to identify all the active effects in a model which significantly deviates from the assumption of effect heredity. Moreover, for screening experiments, missing an active effect is generally considered a major mistake, while a false positive is not considered dramatic. This is because later stages of experimentation will typically reveal that some of the effects initially declared to be active are in fact inactive.

Example 3. Schoen and Mee (2012) described a 48-run experiment carried out at TNO Science and Industry in the Netherlands to identify the best diamond-turning process to polish a mirror. The goal of the 13-factor experiment was to detect the active MEs and 2FI effects on the mean surface roughness of the diamonds. The experimenters used a two-level orthogonal design in which the 13 ME contrast vectors are orthogonal to all 78 2FI contrast vectors and the 2FI contrast vector pairs have absolute correlations of at most 1/3. Mee (2013) analyzed the data for this experiment using the method of Lenth (1989), forward selection and SAMS, and concluded that the MEs of factors A, B, E, G and I and the 2FIs

AD, BE and GI were active.

We re-analyzed the data using the MIO approach with weak heredity constraints for the 2FIs. For illustrative purposes, we set the value of $k_{\max}$ to the number of active effects (according to Mee (2013)) plus 3 and we generated the 10 best models for each $k \in \{1, \ldots, 11\}$. Even though this example requires searching a space of 5.46×10^{13} models, the sequential MIO algorithm only took 31 minutes to find the list of optimal models. We also searched for the best models using the best-subset selection algorithm implemented in JMP v13, because it allows us to impose effect heredity constraints in the model search. However, JMP v13 was unable to deal with models involving eight or more terms when imposing strong heredity. Therefore, our sequential MIO algorithm is computationally more efficient than JMP v13.

Figure 4 shows the raster plot for the 110 models we found. The figure suggests the presence of nine active effects. More specifically, the raster plot shows that the MEs of A, B, E, G and I, and the interactions AD, BE, EH and GI have large estimates (in absolute value) and frequently appear in the best-fitting models. The remaining effects do not show any pattern. Therefore, the MIO approach supports the findings of Mee (2013) and suggests that also the interaction EH is active.

Example 4. Phadke (1986) discussed a router bit experiment involving 32 runs, seven two-level factors and two four-level categorical factors. The goal of the experiment was to find a model that explains the router-bit life. The data from the experiment was analyzed in Wu and Hamada (2009, ch. 7). The two-level factors were labeled using the letters A–C and F–J, whereas the four-level categorical factors were labeled using the letters D and E. In the analysis of Wu and Hamada (2009, ch. 7), factor D was replaced by three two-level contrast vectors labeled D_1, D_2 and D_3. Similarly, factor E was replaced by three two-level contrast vectors labeled E_1, E_2 and E_3. The design used for this experiment was a regular resolution-III design. This means that some MEs are fully aliased with 2FIs; details regarding the aliasing structure of this design are discussed by Wu and Hamada (2009, ch. 7).

Following Phadke (1986), we considered the MEs of all factors and only the 2FIs involving the two-level factors. We use the MIO approach with the strong heredity constraints for the 2FIs in (11) and the grouping constraints in (22) for the MEs of the four-level factors D and E. We generated the 10 best models for each $k \in \{1, \ldots, 14\}$, where, in contrast with the previous examples, $k_{\max}$ was set to $\lfloor n/3 \rfloor + 4$ because of the presence of four-level categorical factors. The sequential MIO algorithm generated this list of optimal models in less than 10 seconds. The corresponding raster plot is shown in Figure 5. The figure shows that the MEs of the factors F, G and J, and the 2FIs GJ and AF are active since they appear consistently in the best models and have large estimates. Other effects that may also be active are the ME of factor B and the 2FIs AG, CJ and FJ. In addition, the estimate corresponding to the contrast vector D_2 is large, which suggests that the four-level categorical factor D is active as well. Note that, whenever D_2 is included in the model, the contrast vectors D_1 and D_3 are also included. The parameter estimates corresponding to D_1 and D_3 are quite large in absolute value too, as witnessed by the rather dark band for these contrast vectors in the raster plot.

The raster plot in Figure 5 suggests that the ME of factor D and the interaction AG are fully aliased because the interaction AG does not appear in the models with the three ME terms corresponding to factor D, and vice versa. Two other effects that, when studying the raster plot, seem fully aliased are the ME of factor B and the interaction AC. The raster plot therefore confirms the aliasing patterns identified by Wu and Hamada (2009, ch. 7). More specifically, these authors point out that the ME of factor B and the interaction AC are fully aliased, and that the parameter corresponding to the contrast vector D_2 is fully aliased with the interaction AG. Therefore, according to the effect hierarchy principle (Wu and Hamada, 2009, ch. 4), it is more likely that the MEs of the factors B and D are active than the 2FIs involving the factor A.

Phadke (1986) declared the MEs of B, D, F, G and J, and the interaction GJ active, while Wu and Hamada (2009, ch. 7) concluded that the MEs of D, G and J, and the interactions GJ and AF were active. The results from our MIO approach confirm all the active effects discovered by Phadke, but they also suggest that the interaction AF is active, which is in line with the results of Wu and Hamada, in addition to the interactions CJ and FJ. Wu and Hamada attributed the influence of AF to a fully aliased interaction involving the categorical factor D and the two-level factor H, which is an interaction effect we did not consider in our analysis. Only a follow-up experiment would allow these two interactions to be de-aliased.

4 A lean MIO approach for automatic model selection

The examples in Section 3 showed that the full MIO approach, involving the sequential algorithm to create the list of best-fitting models, the user-specified search restrictions and the raster plot, possesses Properties 1–3 of a good model selection method and permits the detection of potentially active effects. Therefore, we strongly recommend the use of this full version of the MIO approach when analyzing data from screening designs. In some circumstances (for instance, when an experiment involves many responses), it might be desirable to perform the model selection automatically. This is not our preferred choice as this approach lacks Properties 1 and 2, and therefore fails to recognize aliasing of effects, which is a major concern in the analysis of data from many screening designs.

In the event an automatic model selection is deemed necessary, a lean MIO approach would be to focus on the best-fitting model for each value of the model size parameter k and selecting a final model using an information criterion. In this section, we evaluate this approach using a simulation study involving two-level designs with 7 and 11 factors. We consider MEs and 2FIs, which amounts to 28 and 66 effects for the 7- and 11-factor design, respectively. We compare our automatic selection approach to the Dantzig selector (DS), the LASSO with weak heredity constraints of Bien et al. (2013), and SAMS. We find the approach of Bien et al. (2013) to deal with heredity more appealing than the LASSO extension of Choi et al. (2010) because it guarantees that the LASSO finds an optimal model for any shrinkage degree. We do not consider forward selection because it was outperformed by the DS in the simulation studies of Marley and Woods (2010) and Draguljić et al. (2014). More specifically, both studies showed that, compared to the DS, forward selection tends to miss many active effects when analyzing data from two-level screening experiments.

Our lean MIO approach is based on best-subset selection. To the best of our knowledge, we are the first to compare the performance of best-subset selection to the DS, the LASSO and SAMS for screening experiments using simulations. Hitherto, this was computationally infeasible because solving the best-subset selection problem was computationally demanding. Now, the MIO approach reduces the computational burden of best-subset selection to the extent that its performance can be studied using simulations.

We first detail our simulation protocol and the automatic model selection procedures we compared. Then, we discuss the performance of the automatic selection procedures to correctly identify the active effects.

4.1 Simulation protocol

Our simulation protocol uses two-level designs for 7 and 11 factors with 20 and 40 runs, respectively. More specifically, we use the 7-factor design 20.7.1 and the 11-factor design

40.11.1a from Mee et al. (2017). These designs have ME contrast vectors that are orthogonal to each other and permit the estimation of almost all models including all MEs and up to seven 2FIs. In the 7-factor 20-run design, some 2FI contrast vectors are correlated with other 2FI contrast vectors as well as with ME contrast vectors. More specifically, there are 105 pairs of one ME contrast vector and one 2FI contrast vector with an absolute correlation of 0.2. In addition, there are 99 pairs of 2FI contrast vectors with an absolute correlation of 0.2 and six pairs of 2FI contrast vectors with an absolute correlation of 0.6. In the 11-factor 40-run design, only pairs of 2FI contrast vectors are correlated. More specifically, there are 936 pairs of 2FI contrast vectors with an absolute correlation of 0.2 and 54 pairs of 2FI contrast vectors with an absolute correlation of 0.6 in this design. Therefore, the 7- and 11-factor designs represent the typical screening situation in which certain pairs of effects are highly aliased, others are only aliased to a small extent or yet others are not aliased at all.

For each design, each of our 1,000 simulations consisted of the following steps:

1. We randomly selected the number of active MEs, m, from the set $\{2, 3, 4, 5\}$ and the number of active 2FIs, g, from the set $\{1, 2, 3, 4, 5, 6, 7\}$.

2. We randomly selected m columns from the design matrix and associated these with the m active MEs chosen. Next, we randomly selected g 2FI columns of the model matrix $\mathbf{X}$ subject to the constraint that weak effect heredity is satisfied.

3. We generated the true values, β_u, for the active effects using two scenarios:

    - 'Equal' scenario: The absolute values for the $m + g$ active effects are randomly sampled (with replacement) from the set $\{0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$. A '+' or '−' sign is randomly assigned to each sampled value.
    - 'Unequal' scenario: The absolute values for the m active MEs are randomly sampled (with replacement) from the set $\{2, 2.5, 3, 3.5\}$. The absolute values for the g active 2FIs are randomly sampled (with replacement) from the set $\{0.5, 1, 1.5, 2\}$. A '+' or '−' sign is randomly assigned to each sampled value.

4. We generated a response vector $\mathbf{y}$ using the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is the matrix involving only the contrast vectors corresponding to the active effects, $\boldsymbol{\beta}$ contains the simulated coefficients β_u and $\epsilon_i \sim N(0, 1)$.

5. We performed an automatic model selection using the DS, the LASSO, SAMS and the lean version of our MIO approach.

The ranges of 2–5 active MEs and 1–7 active 2FIs cover a wide range of situations that are likely to occur in practice. In the 'Equal' scenario, the active MEs and 2FIs are assumed to have comparable sizes. In the 'Unequal' scenario, the active MEs are generally larger than the active 2FIs. Mee et al. (2017) used a similar simulation protocol to compare two-level screening designs.

In our simulation, the settings of the automatic model selection procedures were as follows:

- DS and LASSO: We used the automatic selection procedure of Draguljić et al. (2014) discussed in Supplementary Section A.1.3 with values of the tuning parameter t ranging from zero to the largest absolute element of the vector $\mathbf{X}^T\mathbf{y}$. We selected the best model according to the corrected Akaike information criterion:

$$\text{cAIC} = n \log\left(\frac{\text{RSS}}{n}\right) + \frac{2n\tilde{p}}{n - \tilde{p} - 1},$$

  where RSS denotes the residual sum of squares and $\tilde{p}$ denotes the number of nonzero parameters in the model. Next, we computed the OLS estimates of the best model and retained the effects whose estimates have an absolute value larger than $\gamma = 0.5$, the smallest size of an active effect (in absolute value) in the simulations.

- SAMS: We used the settings recommended by Wolters and Bingham (2011). In particular, we used $s_{\max} = 12$ as the maximum number of active effects in the simulations. For each simulated data set, we generated 10,000 models with $k = s_{\max} + 2$ terms and selected the model with the highest entropy.

- MIO: We used Algorithm 1 with $M = 1$ and $k_{\max} = s_{\max}$, and with the weak heredity constraints (12) added to the MIO formulation. As with the DS and the LASSO, we selected the best model according to the cAIC. From this model, we retained the effects whose estimates have an absolute value larger than $\gamma = 0.5$. To limit the computational burden, we imposed a maximum of 60 seconds for the Gurobi solver to search for each model. This means that the MIO models we used in our simulation study may not be optimal. Our preliminary tests showed that the Gurobi solver usually finds the best models with at most 10 terms in less than 60 seconds. For larger models, Gurobi requires a few more minutes to certify optimality.

4.2 Results

We use three measures to compare the automatic model selection methods: sensitivity, false discovery rate (FDR) and type-I error rate. The sensitivity is the proportion of active effects that are successfully detected. The FDR is the proportion of effects declared active that are actually inactive. The type-I error rate is the proportion of inactive effects that are incorrectly declared active. Obviously, the sensitivity should be maximized, while the FDR and the type-I error rate should be minimized. We obtained empirical distributions of these measures for each model selection method using the 1,000 simulations for each combination of design and scenario ('Equal' or 'Unequal'). One of our results is that all model selection methods had type-I error rates below 0.1 for both scenarios with the 7-factor 20-run design, except for the lean MIO approach. For this method, the type-I error rates were below 0.15 with a median of 0.08 for both scenarios. So, the benchmark methods provide slightly smaller type-I error rates than our lean MIO approach for the 7-factor design. Another result was that almost all type-I error rates, including those for the lean MIO approach, were well below 0.05 in both scenarios with the 11-factor 40-run design. Therefore, all model selection methods are comparable when considering the type-I error rate in the 11-factor design.

4.2.1 7-factor design

Figure 6 shows boxplots comparing the distributions of the sensitivity and the FDR of the four model selection methods under investigation for the 7-factor 20-run design. The bottom and top left panels of the figure show the simulation results for the 'Equal' scenario, in which the active effects are obtained from the same distribution. The right panels show the simulation results for the 'Unequal' scenario, in which the active MEs are generally larger than the active 2FIs.

The top left panel of Figure 6 shows that the lean MIO approach outperforms the benchmark methods in terms of the sensitivity for the 7-factor design under the 'Equal' scenario. As a matter of fact, the lean MIO approach is the only one for which the median sensitivity values is as high as 0.9. This means that the lean MIO approach reaches sensitivity values larger than 0.9 more frequently than the benchmark model selection methods. Also, for 75% of the simulated data sets, the lean MIO approach has a sensitivity larger than 0.78. The bottom left panel of Figure 6 shows that the lean MIO approach has a slightly larger FDR than the DS and the LASSO. For more than 75% of the simulated data sets, the FDRs for these three methods are below 0.4. The SAMS method has a zero FDR for nearly all simulated data sets, but it has a smaller sensitivity than the lean MIO approach.

The top right panel of Figure 6 shows that, for 75% of the simulated data sets, the sensitivity values of the lean MIO approach exceed 0.8 for the 7-factor design under the 'Unequal' scenario. In terms of sensitivity, the lean MIO approach outperforms the three benchmark methods, but this comes at the expense of a larger FDR than the benchmark methods; see the bottom right panel of Figure 6. The SAMS method has the lowest FDR for the simulated data sets under the 'Unequal' scenario, since almost all of its FDR values equal zero.

Figure 6 provides a general comparison of the four methods in terms of the sensitivity and FDR for all sizes and numbers of active effects in the simulations. A more detailed comparison of the performance of the methods is given in Figure 7. This figure shows the average sensitivity of the methods as function of the signal-to-noise (SNR) ratio of the active effects, as well as their average FDR as a function of the number of active effects used in the simulated data sets for the 7-factor design. The right and left panels of Figure 7 show the simulation results under the 'Equal' and 'Unequal' scenario, respectively.

The top panels of Figure 7 show that the lean MIO approach generally has a larger average sensitivity than the benchmark methods across all SNRs of the active effects, under both the 'Equal' and 'Unequal' scenarios. For SNRs equal to or larger than 1.5, the lean MIO approach has an average sensitivity of at least 0.9 under both scenarios for the 7-factor design. For a SNR equal to 1, the average sensitivity of this method is equal to 0.8 when the active effects are drawn from the same distribution, and 0.7 when the MEs are generally larger than the 2FIs. The performance of the lean MIO approach in terms of the sensitivity drops when the SNR of the active effects decreases to 0.5. For this SNR value, the average sensitivity of this method is close to 0.4 in the 'Equal' and 'Unequal' scenario. However, for such low SNR value, the average sensitivities of the benchmark methods are even lower.

The bottom panels of Figure 7 show the average FDRs of the four methods for each number of active effects used in the simulations. The lean MIO approach generally has a larger average FDR than the benchmark methods across all numbers of active effects, under both the 'Equal' and 'Unequal' scenarios. For the smallest number of active effects (that is, two active MEs and one active 2FI), the average FDR of the lean MIO approach is as large as 0.4 for the 7-factor design. The average FDR of this method improves as the number of active effects increases. For instance, the best performance of the lean MIO approach in terms of the FDR is when the number of active effects equals the maximum model size considered ($k_{\max}$). More specifically, for true models with 12 effects, this method has an average FDR close to 0.15 in both the 'Equal' and 'Unequal' scenarios. Regarding the benchmark methods, SAMS has average FDR values close to zero across all numbers of active effects, except when there are three, four or five active effects. For these numbers of active effects, the LASSO outperforms all the other methods in terms of FDR.

4.2.2 11-factor design

Figure 8 compares the distributions of the sensitivity and the FDR of the four automatic model selection methods under investigation for the 11-factor 40-run design. The figure's top panels show that the lean MIO approach outperforms the benchmark methods in terms of the sensitivity, under both the 'Equal' and the 'Unequal' scenarios. In both scenarios, the MIO approach has sensitivity values larger than 0.8 for the vast majority of the simulated data sets. Moreover, for more than 50% of the simulated data sets, its sensitivity values equal 1. In both the 'Equal' and the 'Unequal' scenarios, the MIO approach, the DS and the LASSO perform similarly in terms of the FDR. Most of the FDR values for these three methods were smaller than 0.2. As with the 7-factor design, SAMS performs extremely well in terms of the FDR but this good performance is accompanied by a sensitivity smaller than that of the lean MIO approach in both the 'Equal' and the 'Unequal' scenarios.

Figure 9 shows the average sensitivity and FDR as a function of the SNR and number of the active effects, respectively, of the four methods for the 11-factor design. The figure's top panels show that the lean MIO approach generally has a larger average sensitivity than the benchmark methods, across all SNRs of the active effects in the 'Equal' and 'Unequal' scenario. In fact, when the SNR is at least one, the lean MIO approach has an average sensitivity strictly larger than 0.85 in both scenarios. The correct identification of the active effects with a SNR of 0.5 is more challenging as the average sensitivity for the lean MIO approach, as well as for all the benchmark methods, is lower than 0.45. In terms of FDR, the bottom panels of Figure 9 show that the lean MIO approach has a smaller average FDR than SAMS when the number of active effects is equal to three, four or five, in both the 'Equal' and 'Unequal' scenarios. In both scenarios, however, the DS and the LASSO have smaller average FDRs than these methods when the number of active effects is at most six. For seven active effects or more, SAMS consistently has the smallest average FDRs for the 11-factor design under the 'Equal' and 'Unequal' scenario.

4.2.3 Discussion

Overall, the simulation results show that lean version of the MIO approach, in which only the best-fitting model is retained for each k value, is a good strategy to correctly identify active effects as small (in absolute value) as the standard deviation of the noise. The lean MIO approach also outperformed all the benchmark methods in terms of sensitivity. SAMS had lower FDR values than the DS, the LASSO and the lean MIO approach, except when the number of active effects was small. For screening experiments, we prefer a higher sensitivity over a low FDR because it is generally considered more of a problem to miss active effects than to have false positives. This is due to the fact that follow-up experiments will typically reveal that some of the effects initially declared to be active are in fact inactive. Factors that are declared inactive are generally dropped from a study, so that declaring factors to be inactive is irrevocable. Therefore, if an automatic selection of active effects is needed, we recommend the lean MIO approach.

The DS, the LASSO and the lean MIO approach in our simulations involved a threshold $\gamma = 0.5$ to select the active effects and to control the FDRs and type-I error rates. As Phoa et al. (2009), Marley and Woods (2010), Draguljić et al. (2014) and Mee et al. (2017), we used a γ value based on the distributions of the coefficients of the active effects. Therefore, our simulation results, as well as those in the articles cited, may show an overly optimistic performance of the methods. We also studied the performance of these methods for different values of γ ranging from 0 to 1. We found that decreasing the value of γ increases their sensitivity, at the cost of larger FDRs and type-I error rates. On the other hand, increasing the value of γ results in smaller FDRs and type-I error rates, but also smaller sensitivities than those discussed here.

The automatic model selection methods in our simulation study should not be considered a substitute for an expert analysis. In practice, an expert analysis based on the full MIO approach rather than its lean version or any of the other automatic model selection methods will allow a more informed decision on the active effects than any automatic model selection.

5 Concluding remarks

In this article, we proposed a Mixed Integer Optimization (MIO) approach to analyze data from screening designs. The full MIO approach, involving a sequential algorithm to create lists of best-fitting models of different sizes, the user-specified search restrictions and the raster plot, possesses Properties 1, 2 and 3, which are generally considered desirable for a method of analysis. An attractive feature of the MIO approach is that it permits the identification of the potentially active effects while revealing the aliasing of effects due to the screening design used. Our examples in Section 3 showed that model aliasing is a major concern in the analysis of data from screening designs. The MIO approach also provides a flexible framework to take into account subject-matter expertise through linear constraints that can be added to the best-subset selection problem, and it guarantees that the best-fitting models satisfying various user-specified search restrictions will be found. Moreover, the MIO approach renders best-subset regression feasible and relies on least-squares estimation. For all these reasons, the MIO approach should be appealing to practitioners running screening experiments in any branch of science and in industry. Our simulation study demonstrated that a lean, automated version of the MIO approach also has the potential to provide valuable information, since it was able to detect most of the active effects.

The MIO model selection approach relies on the best-subset selection approaches of Bertsimas et al. (2016) and Bertsimas and King (2016), and it greatly benefits from the dramatic improvements in computational hardware and from the theoretical and algorithmic advances in mixed integer optimization in recent years. For instance, the maximum computing time required by the MIO approach to find a complete list of optimal models in Examples 1–4 was 92 min. Other currently available algorithms for best-subset selection are either out of computational reach, do not allow the user to impose constraints on the model search or do not guarantee that the best models are found. In this regard, it was surprising to find that the package 'leaps' v2.9 in R did not find the best models for some of our examples. Therefore, we consider the MIO approach as the only reliable and computationally feasible alternative to perform best-subset selection in screening experiments.

Our implementation of the MIO approach used the solver Gurobi v7.5. Just like the CPLEX and SCIP solvers, the Gurobi solver use state-of-the-art insights from the operations research literature. Future versions of Gurobi will certainly speed up the MIO approach even further. For instance, according to the developers of the solver, Gurobi v8.1 is 2.8 times faster than previous versions for solving mixed integer optimization problems involving a quadratic objective function such as the MIO problem (Gurobi Optimization, Inc., 2019).

A byproduct of the research in this article is that the MIO approach has the potential to analyze data from supersaturated designs. A supersaturated design is a screening design which has more factors than runs (Wu and Hamada, 2009, ch. 9). During the evaluation and validation of the sequential algorithm of the MIO approach, we analyzed the data from the eight two-level, 24-factor 14-run supersaturated designs of Abraham et al. (1999). The data set corresponding to each design was generated by selecting specific sets of 14 rows from the rubber-making experiment data described in Williams (1968). Abraham et al. (1999) analyzed the data from the designs using best-subset selection by generating lists with the best models with up to five effects for each data set. Our sequential MIO algorithm took less than five seconds to find the same lists of models for the eight data sets. Since supersaturated designs possess complex aliasing structures and may involve factors with more than two levels (Yamada and Lin, 1999), we believe that Properties 1–3 of the MIO approach are also desirable to identify the active effects in these designs. A comprehensive evaluation of the MIO approach to analyze data from supersaturated designs is therefore an interesting topic for future research.

In this article, we paid specific attention to model selection methods for screening experiments within the frequentist framework. Within the Bayesian framework, an alternative analysis method is the Bayesian variable selection (BVS) method of Chipman et al. (1997). This method possesses Properties 1–3 of a good model selection method as well, and it has some similarities with the MIO approach. For instance, based on given prior distributions for the effects of interest, the BVS method also decides on the best model using a list of models compatible with the data. Instead of ranking the models in terms of their residual sum of squares, as MIO does, the models are ranked according to their posterior probabilities of being the true model. The BVS method can incorporate user-specified constraints in the model search through the prior distributions of the effects. For instance, Chipman (1996) specifies the required prior distributions of the effects to impose effect heredity, to group the effects of categorical factors and to remove models from the search. So, if the experimenter has a good idea about which prior distributions to use, the BVS method becomes attractive. On the other hand, MIO uses linear constraints that do not require any parameter specification to reflect user knowledge in the model search. A detailed comparison between the MIO approach and the BVS method is another interesting topic for future research.

Supplementary materials

Supplementary sections.pdf Qualitative comparison to benchmark methods.

Programs.zip Python implementation of the MIO approach and examples.

Appendix

In this appendix, we show how to specify the values for the constants B, E, B^L and E^L, in the constraints (7)–(10) of the MIO problem formulation introduced in Section 2.1.2. Bertsimas et al. (2016) showed that the optimal values of $\hat{\beta}_u$ and $\hat{y}_i$ satisfy constraints (7)–(10) when the following bounds are used:

$$B = \tau \beta^{\star},$$

$$B^L = kB,$$

$$E = \left(\max_{i=1,\ldots,n} \|\mathbf{x}_i\|_{1:k} \right) B,$$

and

$$E^L = \min \left\{ B^L \sum_{i=1}^{n} \|\mathbf{x}_i\|_{\infty}, \sqrt{n}\|\mathbf{y}\|_2 \right\}.$$

In these expressions, $\beta^{\star}$ is the smallest absolute $\hat{\beta}_u$ value known to be infeasible, $\tau \geq 1$ is a constant to safeguard against a misspecification of $\beta^{\star}$, $\mathbf{x}_i$ represents the ith row of matrix $\mathbf{X}$, $\|\mathbf{y}\|_2$ denotes the square root of the sum of the squared elements of vector $\mathbf{y}$ (i.e. the l_2-norm of $\mathbf{y}$), $\|\mathbf{x}_i\|_{\infty}$ denotes the largest absolute element of vector $\mathbf{x}_i$ (i.e. the l_{∞}-norm of $\mathbf{x}_i$), and $\|\mathbf{x}_i\|_{1:k}$ is the sum of the k largest absolute entries of $\mathbf{x}_i$. Note that our expression for E^L differs from that in Equation (2.12) in Theorem 2.1 of Bertsimas et al. (2016): $\sqrt{k}$ in their Equation (2.12) needs to be replaced by $\sqrt{n}$ (see their Supplementary Section 8.3).

In the absence of prior information about the value of $\beta^{\star}$, we specify this value using the discrete first-order algorithm in Bertsimas et al. (2016, sec. 3) and use a value of τ equal to two. This τ value is particularly useful to safeguard against a misspecification of $\beta^{\star}$ in the boosting constraints of MIO problems including user-specified constraints, since the discrete first-order algorithm provides an accurate value of $\beta^{\star}$ for the standard MIO problem only.

Acknowledgments

The research that led to this article was financially supported by the Flemish Fund for Scientific Research FWO. Part of the simulation study was carried out at the HPC core facility CalcUA at the University of Antwerp. We are grateful to José Núñez Ares and Nha Vo-Thanh for drawing our attention to the work of Bertsimas, and to Robert Mee for suggesting us to include factor sparsity constraints. We also thank José Núñez Ares for his thoughtful comments on a preliminary version of this article.

References

Abraham, B., Chipman, H., and Vijayan, K. (1999). Some risks in the construction and analysis of supersaturated designs. *Technometrics*, 41:135–141.

Beale, E. M. L. and Forrest, J. J. H. (1976). Global optimization using special ordered sets. *Mathematical Programming*, 10:52–69.

Bertsimas, D. and King, A. (2016). An algorithmic approach to linear regression. *Operations Research*, 64:2–16.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44:813–852.

Bertsimas, D. and Weismantel, R. (2005). *Optimization Over Integers.* Dynamic Ideas Press.

Bien, J., Taylor, J., and Tibshirani, R. (2013). A LASSO for hierarchical interactions. *The Annals of Statistics*, 41:1111–1141.

Bixby, R. (2012). A brief history of linear and mixed-integer programming computation. *Documenta Mathematica. Extra Volume: Optimization Stories*, pages 107–121.

Box, G. E. P. and Meyer, R. D. (1986). An analysis for unreplicated fractional factorials. *Technometrics*, 28:11–18.

Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35:2313–2351.

Cheng, S.-W. and Wu, C. F. J. (2001). Factor screening and response surface exploration. *Statistica Sinica*, 11:553–604.

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 24:17–36.

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

Choi, N., Li, W., and Zhu, J. (2010). Variable selection with the strong heredity constraint and its oracle property. *Journal of the American Statistical Association*, 105:354–364.

Draguljić, D., Woods, D. C., Dean, A. M., Lewis, S. M., and Vine, A.-J. E. (2014). Screening strategies in the presence of interactions. *Technometrics*, 56:1–16.

Errore, A., Jones, B., Li, W., and Nachtsheim, C. J. (2017). Using definitive screening designs to identify active first- and second-order factor effects. *Journal of Quality Technology*, 49:244–264.

Furnival, G. and Wilson, R. (1974). Regression by leaps and bounds. *Technometrics*, 16:499–511.

Gurobi Optimization, Inc. (2019). Gurobi 8 performance benchmarks. Available at http://www.gurobi.com/pdfs/benchmarks.pdf. Accessed 11 July 2019.

Hamada, M. and Wu, C. F. J. (1992). Analysis of designed experiments with complex aliasing. *Journal of Quality Technology*, 24:130–137.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data mining, Inference, and Prediction.* New York: Springer, 2nd edition.

Jones, B. and Nachtsheim, C. J. (2011). A class of three-level designs for definitive screening in the presence of second-order effects. *Journal of Quality Technology*, 43:1–15.

Lenth, R. (1989). Quick and easy analysis of unreplicated experiments. *Technometrics*, 31:467–473.

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

Liu, H., Zhang, J., Jiang, X., and Liu, J. (2010). The group Dantzig selector. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 461–468.

Marley, C. J. and Woods, D. C. (2010). A comparison of design and model selection methods for supersaturated experiments. *Computational Statistics and Data Analysis*, 54:3158–3167.

Mee, R. W. (2013). Tips for analyzing nonregular fractional factorial experiments. *Journal of Quality Technology*, 45:330–349.

Mee, R. W., Schoen, E. D., and Edwards, D. J. (2017). Selecting an orthogonal or nonorthogonal two-level design for screening. *Technometrics*, 59:305–318.

Miller, A. (2002). *Subset Selection in Regression*. Chapman & Hall/CRC, 2nd edition.

Miller, a. and Sitter, R. R. (2001). Using the folded-over 12-run plackett-burman design to consider interactions. *Technometrics*, 43(1):44–55.

Natarajan, B. . (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24:227–324.

Ockuly, R., Weese, M., Smucker, B., Edwards, D. J., and Chang, L. (2017). Response surface experiments: A meta-analysis. *Chemometrics and Intelligent Laboratory Systems*, 164:64–75.

Phadke, M. S. (1986). Design optimization case studies. *AT&T Technical Journal*, 65:51–68.

Phoa, F. K. H., Pan, Y. H., and Xu, H. (2009). Analysis of supersaturated designs via the Dantzig selector. *Journal of Statistical Planning and Inference*, 139:2362–2372.

Schoen, E. D. and Mee, R. W. (2012). Two-level designs of strength 3 and up to 48 runs. *Journal of the Royal Statistical Society Series C*, 61:163–174.

Schoen, E. D., Vo-Thanh, N., and Goos, P. (2017). Two-level orthogonal screening designs with 24, 28, 32, and 36 runs. *Journal of the American Statistical Association*, 112:1354–1369.

Tibshirani, R. (1999). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B*, 52:267–288.

Weese, M. L., Smucker, B. J., and Edwards, D. J. (2015). Searching for powerful supersaturated designs. *Journal of Quality Technology*, 47:66–84.

Westfall, P. H., Young, S. S., and Lin, D. K. J. (1998). Forward selection error control in the analysis of supersaturated designs. *Statistica Sinica*, 8:101–117.

Williams, K. R. (1968). Designed experiments. *Rubber Age*, 100:65–71.

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

Wu, C. F. J. and Hamada, M. S. (2009). *Experiments: Planning, Analysis, and Optimization*. Wiley, 2nd edition.

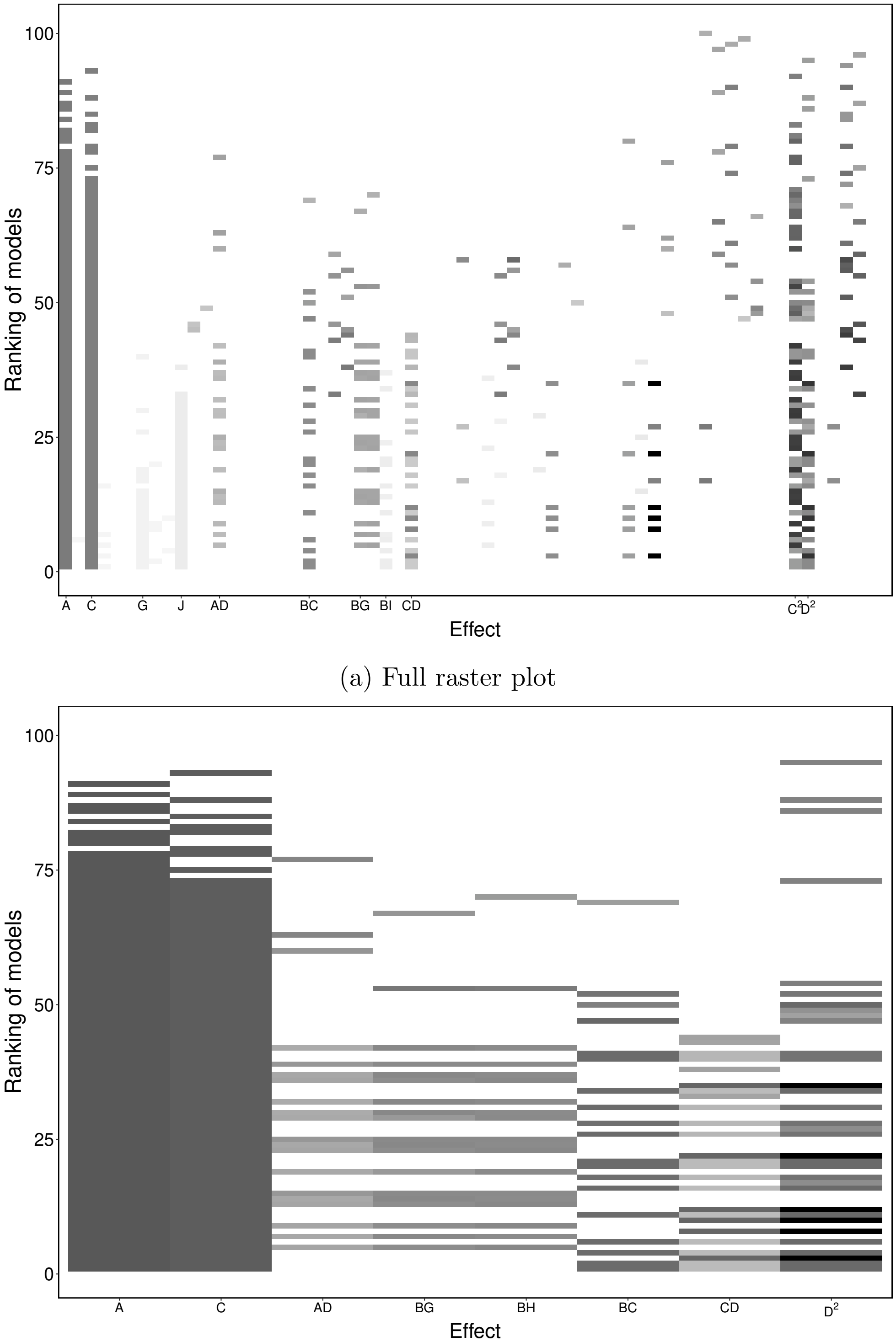
Xu, H., Cheng, S.-W., and Wu, C. (2004). Optimal projective three-level designs for factor screening and interaction detection. *Technometrics*, 46:280–292.

Yamada, S. and Lin, D. K. J. (1999). Three-level supersaturated designs. *Statistics and Probability Letters*, 45:31–39.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67.

Table 1: Design and response vectors for the synthetic experiments in Examples 1 and 2. The response Y_1 was simulated using model (25) while Y_2 was simulated using (26).

Row	A	B	C	D	E	F	G	H	I	J	Y_1	Y_2
1	0	1	1	1	1	1	1	1	1	1	13.61	0.78
2	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	8.89	0.33
3	1	0	−1	−1	−1	−1	1	1	1	1	9.27	0.19
4	−1	0	1	1	1	1	−1	−1	−1	−1	8.86	−2.00
5	1	−1	0	−1	1	1	−1	−1	1	1	5.59	0.59
6	−1	1	0	1	−1	−1	1	1	−1	−1	2.07	−0.72
7	1	−1	−1	0	1	1	1	1	−1	−1	5.28	2.69
8	−1	1	1	0	−1	−1	−1	−1	1	1	5.78	0.84
9	1	−1	1	1	0	−1	−1	1	−1	1	11.62	−1.60
10	−1	1	−1	−1	0	1	1	−1	1	−1	3.29	−4.16
11	1	−1	1	1	−1	0	1	−1	1	−1	10.84	−1.72
12	−1	1	−1	−1	1	0	−1	1	−1	1	2.81	−3.72
13	1	1	−1	1	−1	1	0	−1	−1	1	6.10	0.34
14	−1	−1	1	−1	1	−1	0	1	1	−1	4.32	−2.32
15	1	1	−1	1	1	−1	−1	0	1	−1	5.53	−0.27
16	−1	−1	1	−1	−1	1	1	0	−1	1	6.03	−2.17
17	1	1	1	−1	−1	1	−1	1	0	−1	12.56	4.42
18	−1	−1	−1	1	1	−1	1	−1	0	1	5.80	2.24
19	1	1	1	−1	1	−1	1	−1	−1	0	13.52	4.43
20	−1	−1	−1	1	−1	1	−1	1	1	0	4.59	2.03
21	0	0	0	0	0	0	0	0	0	0	−0.33	−0.18



(a) Full raster plot

(b) Reduced raster plot

Figure 1: Raster plots for the synthetic experiment in Example 1 obtained from standard MIO.

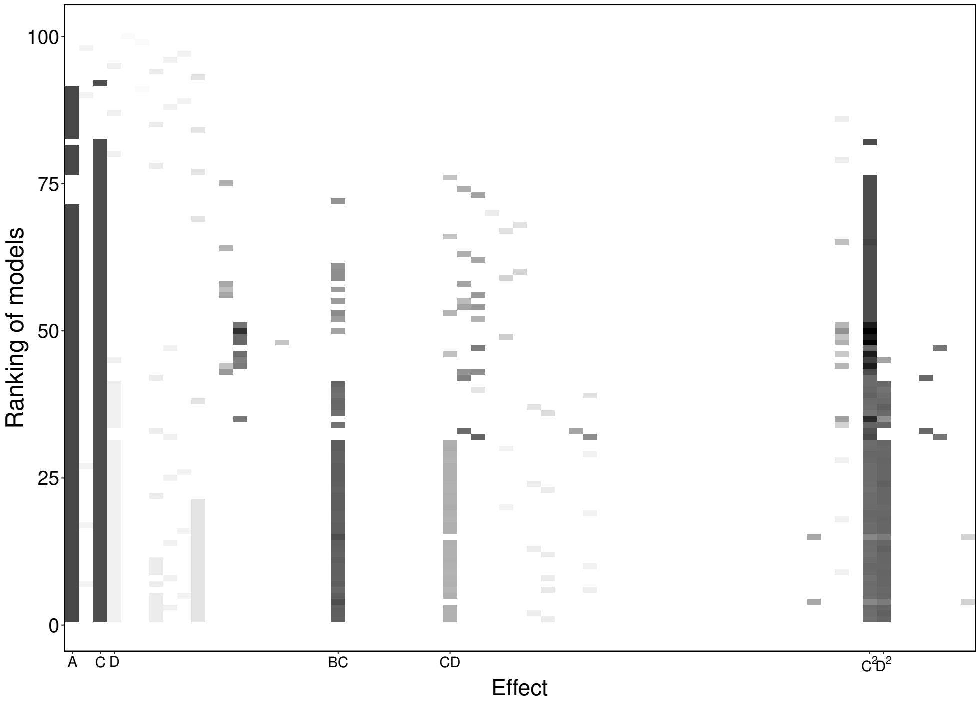


Figure 2: Raster plot for the synthetic experiment in Example 1 obtained from MIO with weak standard and quadratic/interaction heredity constraints for the second-order effects.

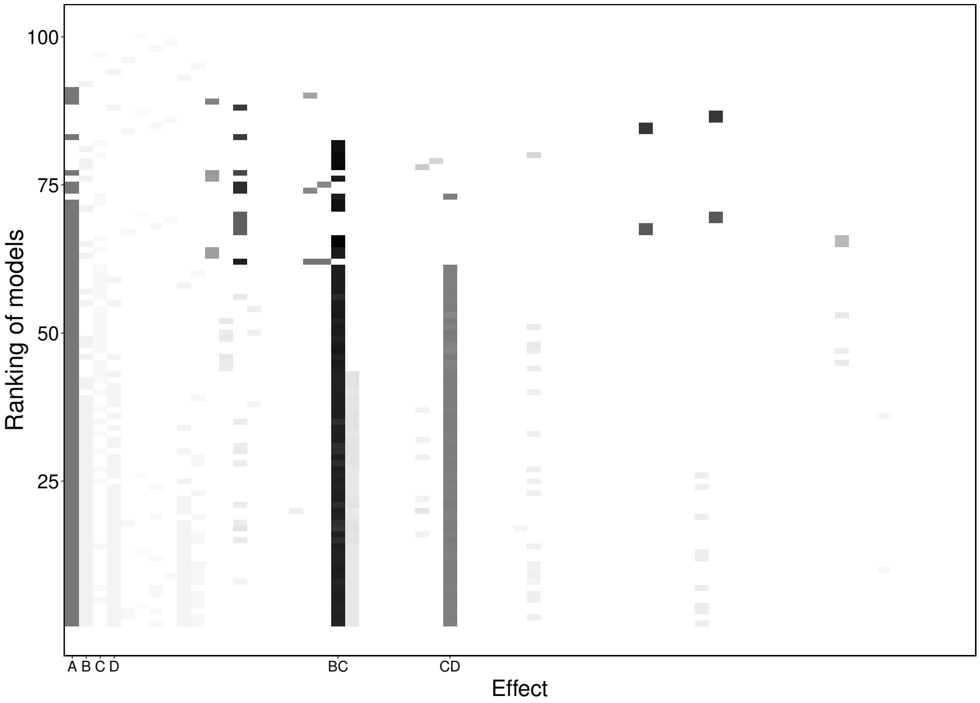


Figure 3: Raster plot for the synthetic experiment in Example 2 obtained from MIO with weak heredity constraints for the second-order effects.

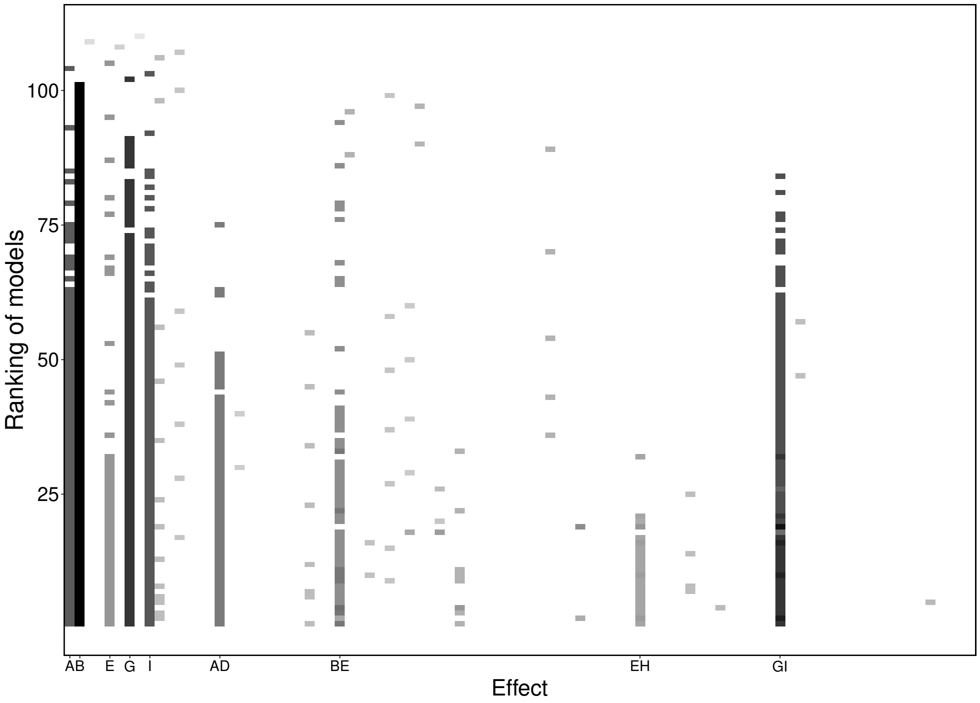


Figure 4: Raster plot obtained from MIO for the mirror-polishing experiment in Example 3. All models follow weak effect heredity.

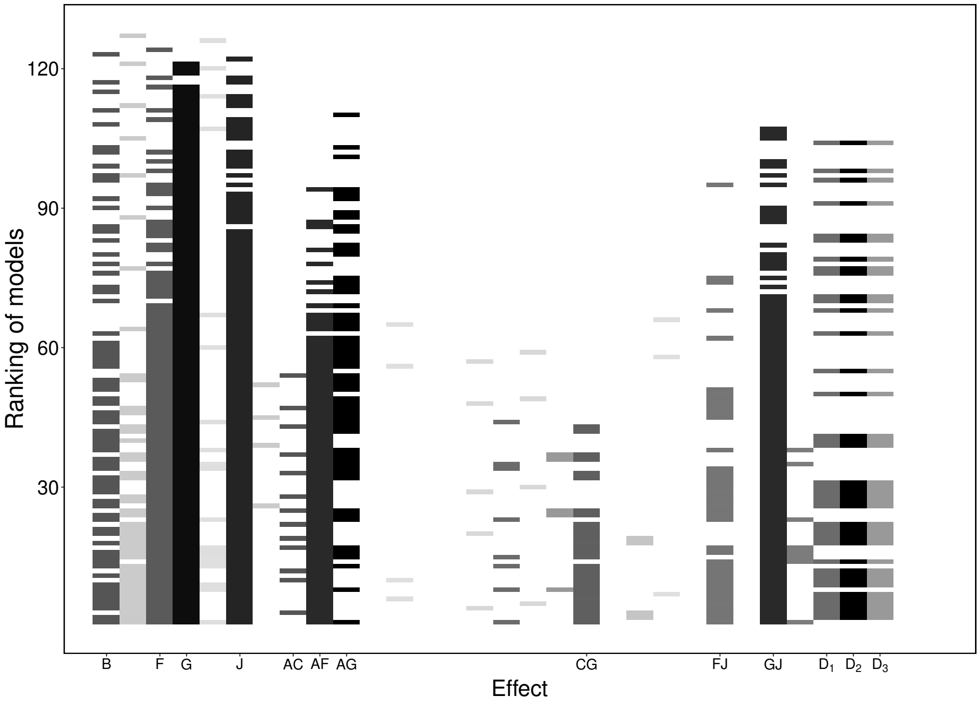


Figure 5: Raster plot obtained from MIO for the router-bit experiment in Example 4. The effects of the four-level factors are shown on the right. All models obey strong effect heredity.

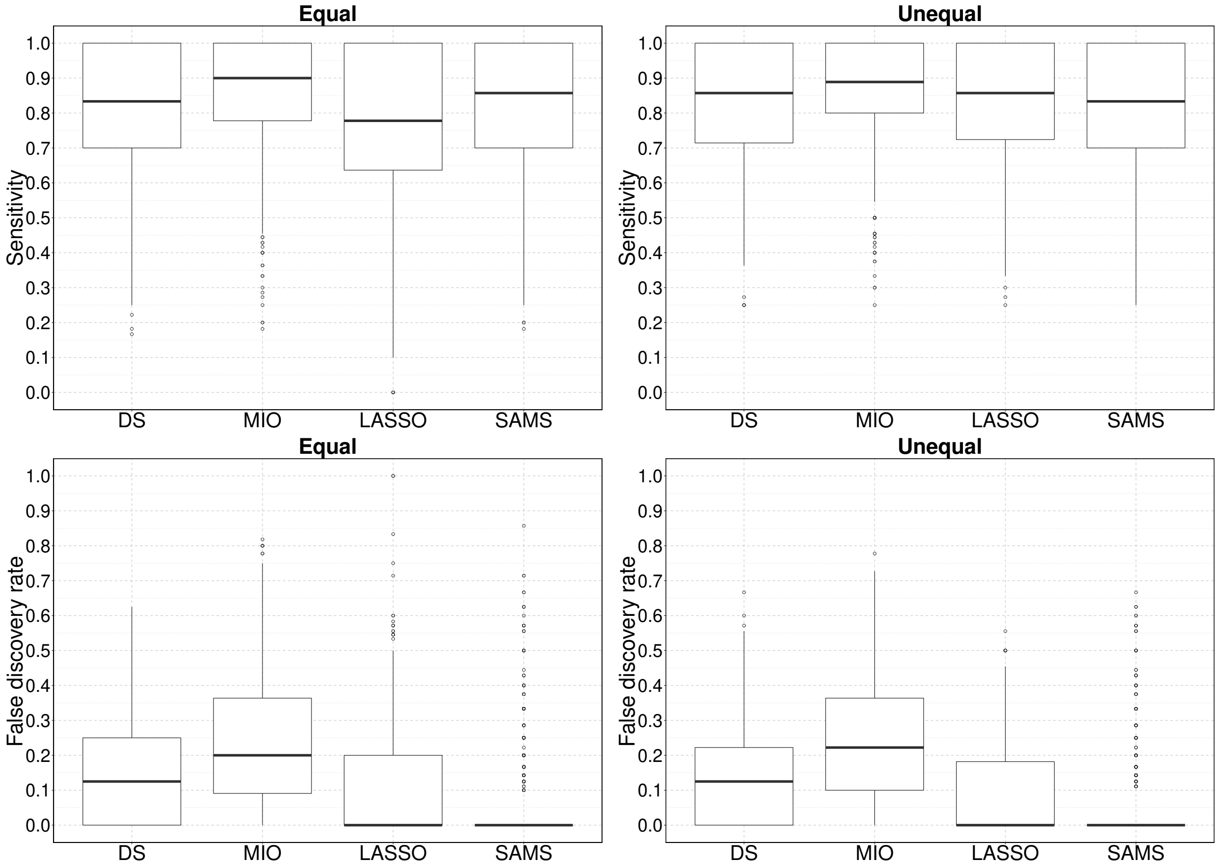


Figure 6: Sensitivities and false discovery rates for four automatic model selection procedures for the 7-factor 20-run orthogonal design. 'Equal' refers to a scenario in which all active effects are obtained from the same distribution, while 'Unequal' refers to a scenario in which active MEs are generally larger than active 2FIs.

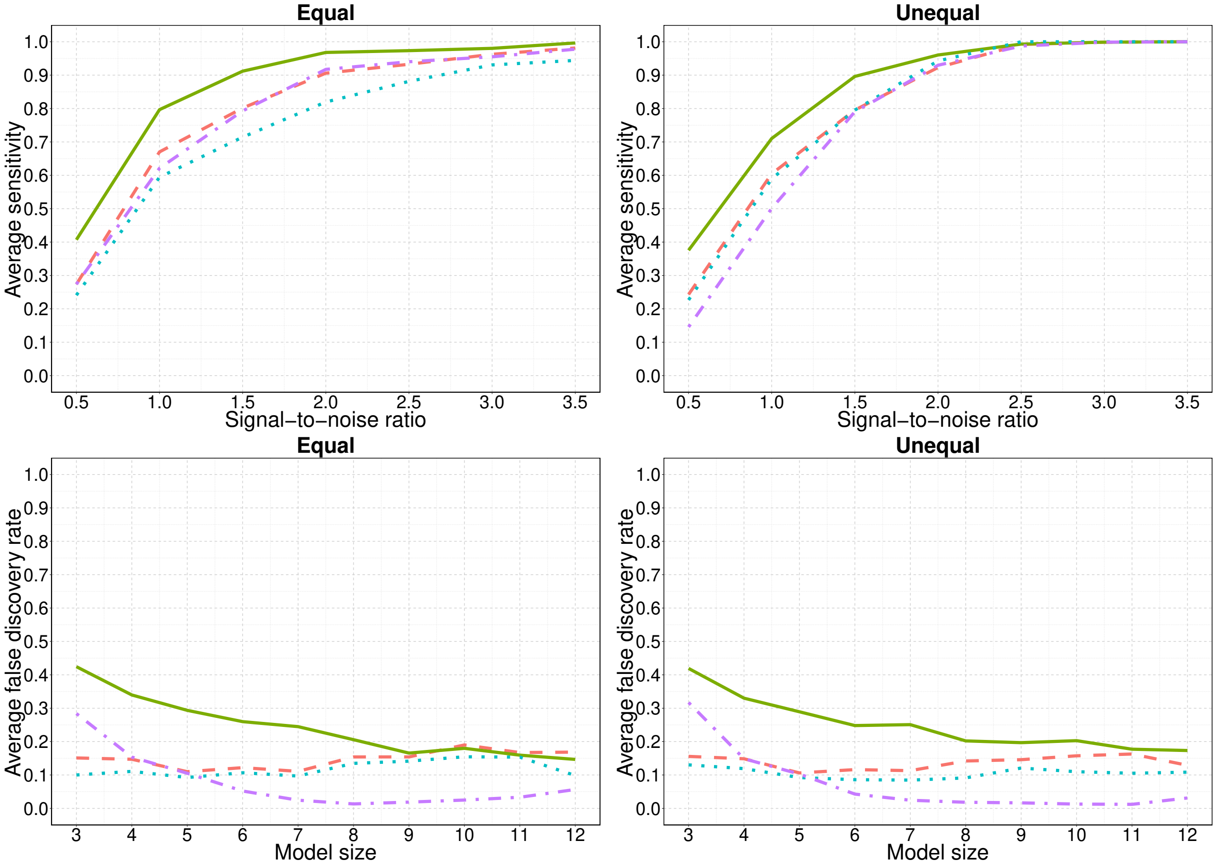


Figure 7: Average sensitivity and false discovery rate for four automatic model selection procedures for the 7-factor 20-run orthogonal design by signal-to-noise ratio of the active effects and by the size of the true model. 'Equal' refers to a scenario in which all active effects are obtained from the same distribution, while 'Unequal' refers to a scenario in which active MEs are generally larger than active 2FIs. MIO: green solid line; DS: pink dashed line; LASSO: blue dotted line; SAMS: purple dash-dotted line. The online version of this figure is in color.

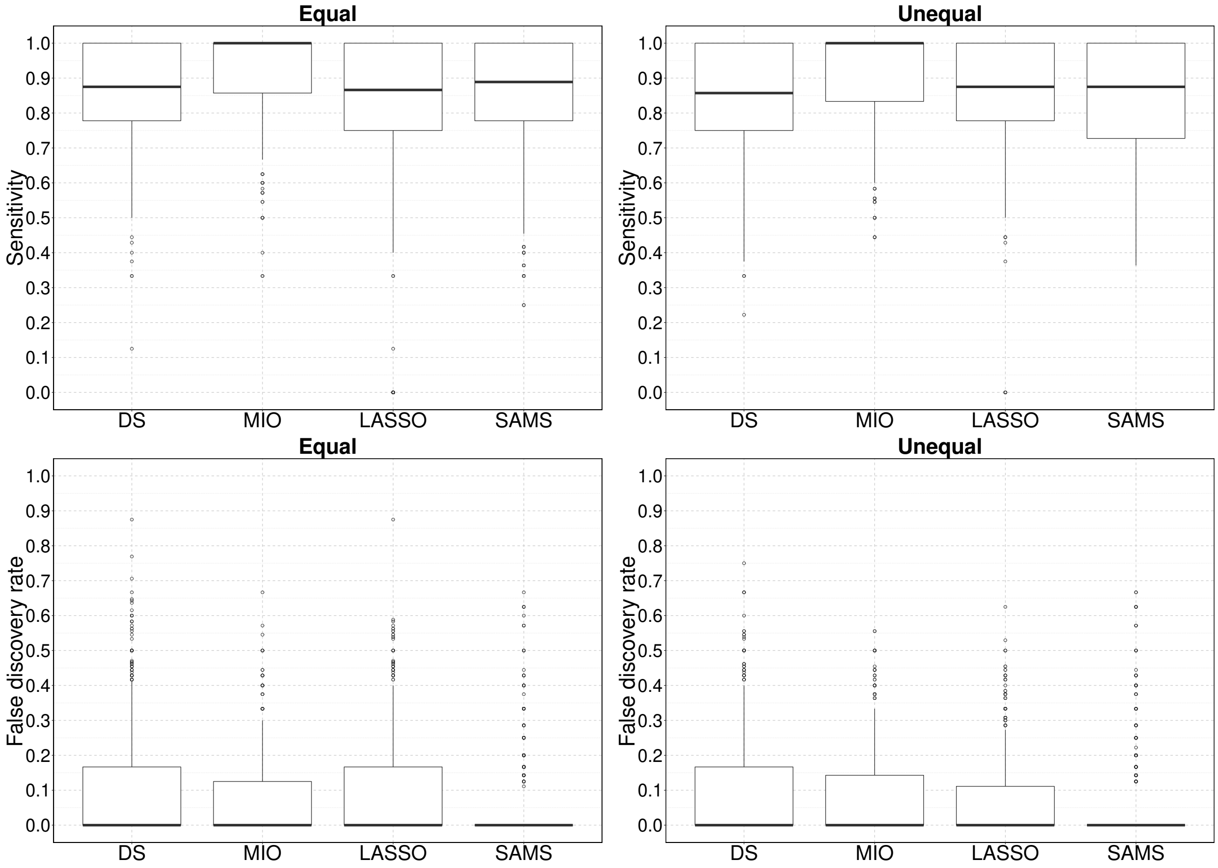


Figure 8: Sensitivities and false discovery rates for four automatic model selection procedures for the 11-factor 40-run orthogonal design. 'Equal' refers to a scenario in which all active effects are obtained from the same distribution, while 'Unequal' refers to a scenario in which active MEs are generally larger than active 2FIs.

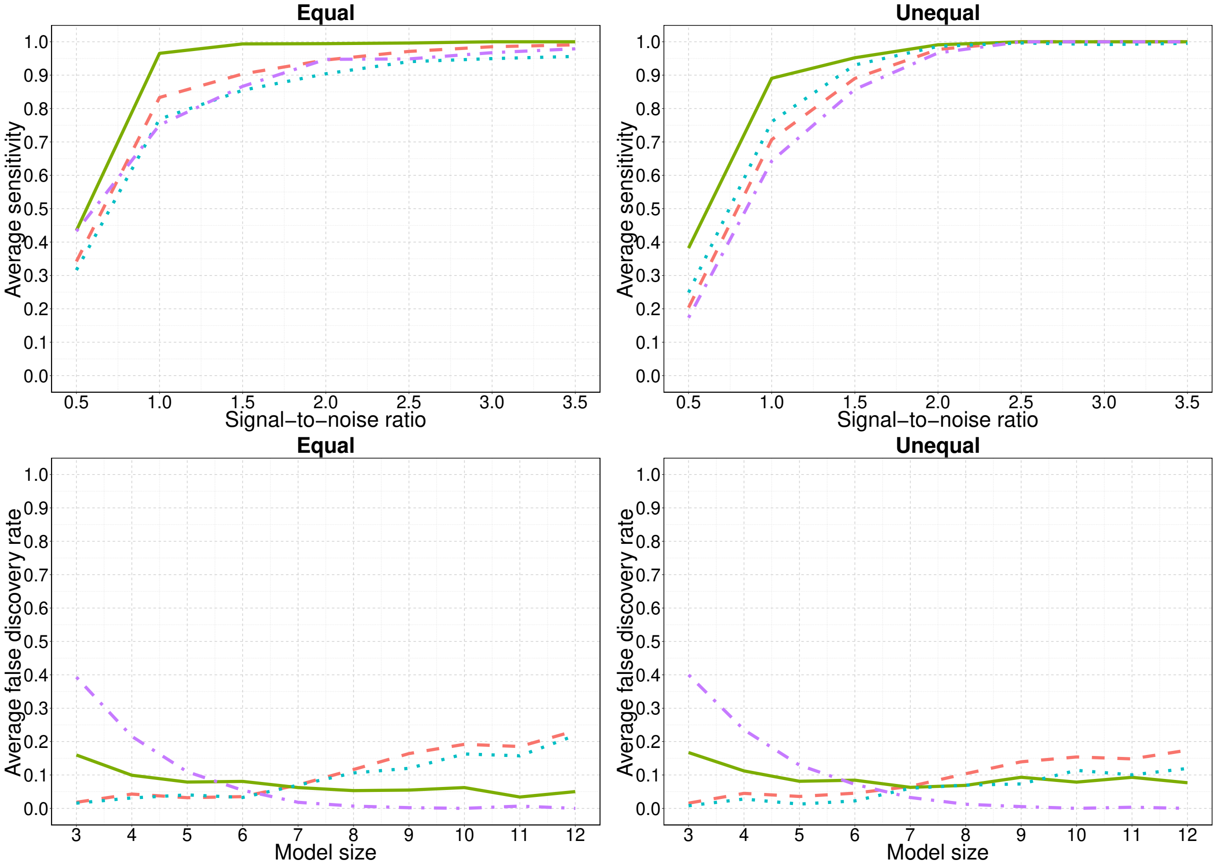


Figure 9: Average sensitivity and false discovery rate for four automatic model selection procedures for the 11-factor 40-run orthogonal design by signal-to-noise ratio of the active effects and by the size of the true model. ‘Equal’ refers to a scenario in which all active effects are obtained from the same distribution, while ‘Unequal’ refers to a scenario in which active MEs are generally larger than active 2FIs. MIO: green solid line; DS: pink dashed line; LASSO: blue dotted line; SAMS: purple dash-dotted line. The online version of this figure is in color.