Model-robust design of mixture experiments

Paul Kristoffersen[1] and Byran J. Smucker[2*]

[1]Department of Mathematics

Miami University, Oxford, OH 45056

[2]Department of Statistics

Miami University, Oxford, OH 45056

Abstract

Optimal designs are often used for constrained mixture experiments because of the irregular design spaces. For these experiments, the number of blends needed to fit standard linear models may be too large when considering second- or third-order terms. We present a computationally-tractable algorithm for generating model-robust mixture designs that exploits anticipated effect sparsity by using a set of models defined by a user-specified number of higher-order terms. We compared the model-robust designs with Bayesian-optimal designs, and the model-robust designs show an improved ability to either estimate realistic models or make predictions for mixture experiments.

Keywords: optimal design; mixture experiments; robust design; D-optimality; I-optimality

Introduction

The goal of this paper is to present an algorithm for generating model-robust designs for constrained mixture experiments. This approach provides the experimenter with the ability to estimate high-order mixture terms even for experiments with a large number of components when resources are limited.

Mixture experiments are widely used in fields where product formulation is required. Smith (2005) provides a list of more than 30 categories of such products, including beverages, cosmetics, floor coverings,

*corresponding author, smuckerb@miamioh.edu

glasses, paints, and pesticides. These experimental settings are as ubiquitous as the chemical products that provide the largely unnoticed backdrop of our lives: polymers, dyes, pharmaceuticals, sealants, and many more.

Mixture experiments are challenging to design because the components of the mixture cannot be independently varied. That is, traditional mixture experiments are constrained by the requirement that the sum of the component proportions is one, which results in a simplex-shaped design region. Complicating the problem further is that there are often individual and multicomponent constraints that make the design space irregularly shaped. Because of the unique design regions, classical mixture designs such as the simplex-lattice design (see, e.g., Cornell 2002; Smith 2005) often fail to provide the necessary flexibility. Consequently, optimal designs enjoy widespread use within the mixture experiment literature. There are several optimality criteria commonly used when generating optimal designs. We focus on D- and I-optimality in this paper: the D-criterion provides precise estimation of coefficients in the linear regression model, while the I-criterion minimizes the average variance of prediction across the design space.

Optimal designs require, among other things, specifying a model form relating the response to the components. Consequently, an optimal design is only optimal with respect to the assumed model. If the specified model is different from the true model, the generated design will not be optimal and may not even be able to estimate the parameters β in the linear model of the response, $Y = X\beta + \epsilon$. It is this limitation that model-robust designs address.

Literature review

Several model-robust experimental design approaches have been suggested in the literature. For instance, Welch (1983) and Fang and Wiens (2000) construct**ed** designs that guard against departures from the assumed model form using mean-squared error ideas, and Gutiérrez and Martın-Martın (2016) provide**d** a version of the same approach in the context of mixture experiments. DuMouchel and Jones (1994) provide**d** a Bayesian design framework that allows protection against potential model terms that may or may not be necessary to fit. This latter approach is the most similar to ours because it allows the experimenter to specify a full model that is larger than the number of experimental runs. The procedure specifies some terms as primary, and these must be estimable. Other terms are deemed potential, and some estimability is maintained for them. Though **Bayesian-optimal designs require specifying** a tuning parameter (the prior variance for the potential terms), we compare these designs to the designs we propose.

The basic model-robustness idea we use is a simple one: generate designs to be optimal for a set of possible models that may be fit. Läuter (1974), Cook and Nachtsheim (1982), Dette (1990), and Dette

and Franke (2001) used this approach in non-mixture settings while assuming asymptotically large designs. Other researchers studied more practical set-of-model approaches that do not require the number of runs to be greater than the number of terms in the specified maximal model (Li and Nachtsheim 2000; Loeppky et al. 2007; Jones et al. 2009; Smucker et al. 2012). The sizes of these model sets tend to grow quite large, so that computational methods to construct the designs are intractable. Smucker and Drew (2015) proposed a method that scales computationally for large model spaces, by making the design robust to a representative sample of models from the model space. Tsai and Gilmour (2010) presented a different computationally efficient algorithm for finding model-robust designs related to the A-criterion. However, neither the algorithms of Smucker and Drew (2015) nor Tsai and Gilmour (2010) can construct model-robust mixture experiment designs straightforwardly.

The literature combining model-robustness and mixture experiments is fairly sparse. Heredia-Langner et al. (2004) presented a genetic algorithm that is able to generate model-robust designs for mixture experiments using the set-of-models approach and Smucker et al. (2011) developed an exchange algorithm to accomplish the same thing. Limmun et al. (2018) constructed model-robust I-optimal mixture designs using a weighted-optimality approach. These papers all considered relatively small mixture experiments, however, with only a small number of possible models. They also tend**ed** to consider model spaces that **were** composed of a set of submodels of a specified maximal model. The downside of these model spaces is that they require sufficient resources to estimate the full model. In resource-constrained settings, especially for larger mixture experiments, this requirement is unrealistic.

As mentioned above, optimal design methods are often used to design mixture experiments because of irregularly-shaped design regions. Because of the mixture constraint, that $\sum_{i=1}^{q} x_q = 1, x_i \geq 0 \; \forall i$ for a mixture design with q factors $(x_1, ..., x_q)$, traditional coordinate exchange algorithms (e.g. Meyer and Nachtsheim 1995) cannot be used to generate optimal designs. Alternatives have been presented, including by Piepel et al. (2005) and Gotwalt and Lancaster (personal communication), though neither of these procedures are model-robust. This latter approach has been implemented in JMP, and a version of this heuristic is the engine we build our model-robust algorithm upon.

In summary, though the literature is replete with model-robust methods for non-mixture experiments, there are few approaches that can accommodate mixture components and irregularly-shaped design regions. Also, little has been developed to allow estimating some higher-order blending terms in settings with limited resources. The Bayesian approach of DuMouchel and Jones (1994) is a notable exception, and we compare our designs with these Bayesian optimal designs.

Motivating examples

Most mixture experiments have only a few components, in which case a reasonably sized design is adequate to fit the usual second-order or cubic models (see Section). However, there are many larger examples in the literature as well. With six components (e.g. experiments described in Piepel (1983) and Smith (2005)), the standard second-order Scheffé model requires at least 21 blends, and to fit the special-cubic model requires at least 41 **blends**. Piepel (1982) reports an 11-component mixture experiment. To fit a special-cubic model would require at least 231 design points, and even a second-order Scheffé model requires at least 66. Even larger mixture experiments have been discussed in the literature, including one with 12 components (Fleury et al. 2014) and one with 21 (Piepel et al. 2005). The methodology developed in this article allows the experimenter to specify the number of terms of a particular type (e.g. quadratic or special cubic) that must be estimated without requiring that all of them are estimable.

To take the most extreme case, consider in more detail the 21-component mixture experiment described by Piepel et al. (2005). The setting was quite complex. Before the experiment was designed, 144 existing blends already existed, and the goal was to augment them with 45 new runs. Instead of using a particular optimality for all of the new runs directly, the designers in Piepel et al. (2005) used a layered approach (Piepel et al. 1993). The new design was generated in 4 stages: 8 outer layer runs, 27 inner layer runs, a centroid run and 4 replicates, and lastly, 5 blends including all 21 components. Except for the centroid and replicates, all of the new runs were chosen using the D-criterion with a linear model. Another complexity: The first 40 runs were chosen to only include 19 of the components while the last 5 included all 21.

Rather than including all of the complexities of the original example, we address the simplified problem of generating a 45-run design from scratch including all 21 components. Since the quadratic model has more than 230 terms and the special cubic more than 1,500, in typical circumstances an optimal design for either of these full models is unrealistic. Instead, we can use a model-robust design that assumes sparsity in the quadratic and special-cubic terms and thus allows some estimability of those terms, while still capping the design at 45 runs. We revisit this design problem in the Empirical results section.

This article continues as follows. In the Background section, we formally define the mixture design problem. Because Gotwalt and Lancaster (personal communication) has not been published in the literature to our knowledge, we then present the basic optimal design algorithm in the "Construction of model-robust mixture designs" section, along with the model-robust elaboration. We demonstrate and discuss the effectiveness of the model-robust designs in the "Empirical results" section, while also considering limitations and challenges. Finally, we end with a discussion and conclusion.

Background

Suppose the mixture experiment of interest requires a design $\xi_n = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T$ of n runs in q components. The usual linear regression model $y_i = f^T(\mathbf{x}_i)\beta + \epsilon_i$ is assumed, where f is a $p \times 1$ model function which specifies the terms included in the model, and $\boldsymbol{\beta}$ is the associated $p \times 1$ parameter vector. In matrix form, this is $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{y}$ and $\boldsymbol{\epsilon}$ are $n \times 1$ vectors and the ϵ_i are iid with $\epsilon_i \sim N(0, \sigma^2)$. The information matrix, $(X^T X)/\sigma^2$, is the inverse of the variance-covariance matrix of $\hat{\beta}$, where $X = [f(\mathbf{x}_1), ..., f(\mathbf{x}_n)]^T$ is ξ_n expanded according to f.

Optimal designs are generated with respect to a criterion which determines the objective function for the associated optimization problem. Two common optimality criteria along with their respective objective functions follow in the "Optimality criteria" section. These criteria are chosen because of their relation to relevant statistical considerations.

Common mixture models

The most common models for mixture experiments were developed by Scheffé (1958). The first-order Scheffé model is $Y = \sum_{i=1}^{q} \beta_i x_i + \epsilon$. This model does not contain a constant term because it would cause a linear dependency among the terms due to the mixture constraint. By substituting $\sum_{i=1}^{q} x_i = 1$, the first-order Scheffé model can easily be derived from a standard first-order linear regression model. Through similar substitutions, the higher order Scheffé models can be derived. The second-order Scheffé model is

$$Y = \sum_{i=1}^{q} \beta_i x_i + \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} \beta_{ij} x_i x_j + \epsilon,$$

and the special cubic model is

$$Y = \sum_{i=1}^{q} \beta_i x_i + \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} \beta_{ij} x_i x_j + \sum_{i=1}^{q-2} \sum_{j=i+1}^{q-1} \sum_{l=j+1}^{q} \beta_{ijl} x_i x_j x_l + \epsilon.$$

We note that Piepel et al. (2002) and Smith (2005) have discussed ~~so-called~~ partial quadratic mixture models**,** which include a subset of the cross-product quadratic terms along with squared quadratic terms. Becker (1968) also suggested alternative mixture models. Though these models fit within the framework of the methodology discussed in this article, studying them is beyond the scope of this work.

Optimality criteria

We consider two optimality criteria (D-optimality and I-optimality) in this article. In the "Construction of model-robust mixture designs" section, we develop model-robust versions of these criteria.

D-optimality

A D-optimal design maximizes the determinant of the information matrix, and produces relatively precise parameter estimates and relatively small correlations between parameter estimates. Consider the optimization problem

$$\begin{aligned} \text{Minimize} \quad & d_D(\xi) = -log(det(\sum_{i=1}^{n} f(\mathbf{x}_i) f^T(\mathbf{x}_i))) \\ \text{Subject to} \quad & A\mathbf{x}_i \leq \mathbf{b} \quad i = 1, ..., n \\ & A_{eq}\mathbf{x}_i = \mathbf{b}_{eq} \quad i = 1, ...n \end{aligned}$$

Here we let $\sum_{i=1}^{n} f(\mathbf{x}_i) f^T(\mathbf{x}_i) = X^T X$. Note also that $A_{eq} = \mathbf{1}_{1\times q}$ and $b_{eq} = 1$, constituting the mixture constraint.

The algorithm we describe in Section utilizes gradient information. For the D-optimality objective, the derivative of the objective function with respect to the jth element of the ith mixture is given by Gotwalt and Lancaster (personal communication) as

$$\frac{\partial d_D(\xi)}{x_{ij}} = -2 f(\mathbf{x}_i)^T (X^T X)^{-1} \frac{\partial f(\mathbf{x}_i)}{\partial x_{ij}}.$$

Then the gradient of the objective function with respect to row i is the vector whose jth element is the above derivative. We provide a derivation in Appendix A.

I-optimality

The I-optimality criterion minimizes the average variance of the predicted response over the design region. This is useful here because mixture experiments are often more concerned with prediction than parameter estimation. The I-optimal optimization problem has the same form as the D-optimal except that the objective function is

$$d_I(\xi) = \int_R f^T(\mathbf{x})(X^T X)^{-1} f(\mathbf{x}) d\mathbf{x}.$$

For ease of computation, this criterion can equivalently be calculated as $d_I(\xi) = trace((X^TX)^{-1}M)$ (Goos and Jones 2011). Here X is as before and M is a moments matrix calculated as $M = \int_R f(\mathbf{x})f^T(\mathbf{x})d\mu_{\mathbf{x}}$. R is the feasible region defined by the linear equality and inequality constraints and $\mu_{\mathbf{x}}$ is a measure defined on R. For the purposes of this paper, $\mu_{\mathbf{x}}$ will always be the uniform distribution.

The gradient of the I-optimal objective function is calculated similarly to the D-optimal one. In this case, the derivative of the objective function $d_I(\xi) = trace((X^TX)^{-1}M)$ with respect to the jth element of the ith row is

$$\frac{\partial d_I(\xi)}{x_{ij}} = -2f(\mathbf{x}_i)^T(X^TX)^{-1}M(X^TX)^{-1}\frac{\partial f(\mathbf{x}_i)}{\partial x_{ij}}.$$

The derivation of this equation is given in Appendix A.

Computing the moments matrix M can pose some difficulty. If the design region is irregular, the integral is decidedly nontrivial. However, Monte Carlo integration provides a straightforward method to estimate the integral. Monte Carlo integration estimates the integral $N(g) = \int_R g(x)dx$ as $N(g) \approx \frac{V}{n}\sum_{i=1}^{n} g(x_i)$ where V is the volume of the region R, and the points x_i are a random sample used to estimate the mean of the function $g(x)$ over R. The next section describes an algorithm used to generate random initial designs. This algorithm can also be used to choose a uniform random sample of points for this Monte Carlo integration. In order to calculate the volume of the feasible region, a formula described by Crosier (1986) is used when there are only upper and lower bounds on the component proportions. If there are multicomponent constraints, the volume must be estimated by sampling from the region defined by the upper and lower bound constraints which has known volume. The proportion of feasible points in the sample estimates the proportion of the known volume in the feasible region.

Construction of model-robust mixture designs

In this section we provide a description of the algorithm and its main aspects.

Model-robust criteria

Suppose $f_1, f_2, ..., f_m$ are m different model functions that could reasonably specify the relationship between the experimental factors (mixture components) and the response. To generate model-robust designs, we simply optimize with respect to the mean of the individual model objective functions. That is, the model-

robust D-optimal problem is

$$\begin{aligned} \text{Minimize} \quad & d_{DM}(\xi) = \frac{1}{m}\sum_{j=1}^{m} -log(det(X_j^T X_j)) \\ \text{Subject to} \quad & A\mathbf{x}_i \leq \mathbf{b} \quad \forall i = 1, ...n \\ & A_{eq}\mathbf{x}_i = \mathbf{b}_{eq} \quad \forall i = 1, ...n \end{aligned}$$

where $X_j^T X_j = \sum_{k=1}^{n} f_j(\mathbf{x}_k) f_j^T(\mathbf{x}_k)$. Similarly, for the I-optimal problem, use the mean of the objective function $trace((X_j' X_j)^{-1} M)$ evaluated over each of the m models. The gradient for the model robust objective will then be the mean of the gradients for the single model objectives. That is, for the model-robust D-optimal case, the derivative of the objective function with respect to the jth element of the ith design row is

$$\frac{\partial d_{DM}(\xi)}{x_{ij}} = \frac{1}{m}\sum_{j=1}^{m} -2 f_j(\mathbf{x}_i)^T (X_j^T X_j)^{-1} \frac{\partial f_j(\mathbf{x}_i)}{\partial x_{ij}}.$$

Then, the gradient with respect to row i is the vector whose jth element is the above derivative. Similarly, the gradient of the model-robust I-optimality objective function is a vector whose elements are of the form

$$\frac{\partial d_{IM}(\xi)}{x_{ij}} = \frac{1}{m}\sum_{j=1}^{m} -2 f_j(\mathbf{x}_i)^T (X_j^T X_j)^{-1} M_j (X_j^T X_j)^{-1} \frac{\partial f_j(\mathbf{x}_i)}{\partial x_{ij}}.$$

Algorithm for model-robust mixture designs

The algorithm we present utilizes MATLAB's built-in nonlinear optimization function *fmincon* (MAT 2013) in the context of the heuristic of Gotwalt and Lancaster (personal communication). Note that our approach does not depend upon the particular nonlinear optimization implementation, and indeed, performance could likely be improved by using dedicated optimization software like CPLEX (CPLEX 2016) or Gurobi (Gurobi Optimization 2016). A description of the algorithm follows, along with a discussion of the procedure used to generate initial designs.

Algorithm description

The algorithm loops through the rows (design points) of the current mixture design and considers replacing each one with a different blend that induces the greatest reduction in the objective function. This is done by applying *fmincon* to the individual row while holding the rest of the design fixed. This makes the algorithm similar to a row-exchange algorithm (Fedorov 1972; Cook and Nachtsheim 1980) since the rows are optimized

individually and the objective function is checked for improvement. On the other hand, it is unlike a classical row-exchange algorithm since no candidate list of points is required. If there is no improvement in the objective function since the last iteration through each of the rows, the algorithm terminates and the current design is returned.

Using *fmincon* to replace the current row is one of the places where our implementation differs from that of Gotwalt and Lancaster (personal communication). In their algorithm, a feasible descent direction is generated using Wolfe's Reduced Gradient Method (WRG) (Bazaraa et al. 2006). After finding this direction, a line search is performed to choose the new mixture to replace the current row. In our version, a row replacement is done in one step by giving the simplified optimization problem to *fmincon*. We found that the WRG approach included a nondegeneracy assumption that was not always satisfied by the mixture experiments we considered. We also explored the Gradient Projection Method of Rosen (Bazaraa et al. 2006), which performed poorly in comparison to *fmincon*.

In the context of generating mixture designs, the inputs to *fmincon* are an objective function, an initial feasible point, matrices of coefficients for the linear inequality and equality constraints, the constant vectors associated with the constraint matrices, and a choice of internal algorithm. Of the several options in *fmincon*, we used the internal MATLAB 'active-set' algorithm. It provided the best results and was also the quickest among the options. The active-set algorithm is essentially a sequential quadratic programming (SQP) procedure. That is, it uses a series of simpler quadratic programs to estimate the Lagrange multipliers for the Karush-Kuhn-Tucker conditions.

Note that the proposed algorithm has the same structure for all of the different optimality criteria discussed earlier. The only difference between them is the choice of objective function. Also, as with other heuristics of this type, there is no guarantee of finding a global optimum. In order to improve the likelihood of finding a high-quality solution, it is best to run the algorithm many times from different random starting designs (see the "Initial designs" section below), then compare the objective function values to select the best one. An outline of the procedure follows.

Algorithm Pseudocode

Generate an initial feasible design ξ (as in "Initial designs" section below)
while true **do**
  for i=1:N **do**
    Replace row i with new row providing greatest improvement in objective function.
  end for
  if No improvement in objective function **then**

break

end if

end while

Return current design.

We also note that an alternative to the algorithm we describe would be to provide the entire optimization problem to *fmincon*, rather than providing it just one row at a time. However, the results when we explored this approach were quite poor in comparison to those generated by the algorithm in this section.

Initial designs

Our algorithm requires a method for generating an initial feasible design, which is **challenging** in highly constrained mixture design spaces. We used the method described in Fang and Yang (2000) and Borkowski and Piepel (2009) to generate approximately uniformly distributed random feasible designs. The method is a variation on simple rejection sampling. First, a uniformly distributed point is generated that satisfies the mixture constraint and all upper and lower bound constraints. This point is generated by sequentially sampling coordinates $1, ..., q-1$. The distribution for each coordinate depends upon the previously sampled coordinates, to maintain the desired uniformity. The last coordinate, q, is chosen so that the sum is equal to one. Once the point is generated, it is then checked to see if it also satisfies the multi-component constraints. If not, the point is rejected and a new point is generated. This process is repeated until the required number of rows for the initial design are generated.

The arbitrary mixture terms model space and its approximation

If we use the model-robust D- or I-criterion described previously, we must specify a set of models upon which the algorithm can operate. The model space we consider in this work is the set of models with a user-specified number of terms of each type from the larger Scheffé models. We call it the Arbitrary Mixture Terms (AMT) model space and denote this set as $\mathcal{F}$ with cardinality m (that is, $\mathcal{F}$ is a collection of m models). Note that mixture models necessarily include all first-order terms $x_1, ..., x_q$. In our examples, the possible higher-order terms of interest are second-order terms $x_i x_j$ and third-order terms $x_i x_j x_l$, which are found in the Scheffé special-cubic model. For each type of higher-order term, the user specifies the expected number of active terms (or an expected upper bound on this quantity). We let g_2 and g_3 be the specified number of second-order and third-order terms respectively. While we did not consider terms from models larger than the Sheffé special cubic, the same idea could be easily extended for higher-order terms.

As an example, consider a mixture experiment with $q = 6$ components. Then there are $\binom{6}{2} = 15$ second

order terms. Under the effect sparsity principle (Li et al. 2006; Ockuly et al. 2017), perhaps the experimenter expects no more than $g_2 = 5$ of the 15 second-order terms to be important. Assuming terms higher than second-order are not of interest for this experiment, $\mathcal{F}$ is the set of models with all 6 components and 5 of 15 second-order blending terms. Then the number of models in $\mathcal{F}$ is $m = \binom{15}{g_2} = \binom{15}{5} = 3003$. On the other hand, if the experimenter decides to allow for up to 3 third-order terms—of which there are $\binom{6}{3} = 20$—then the number of models in $\mathcal{F}$ swells to $m = \binom{15}{5}\binom{20}{3} = 3{,}423{,}420$.

As seen from the simple example above, the AMT model space quickly grows quite large, even as the the number of components and number of expected active model terms grow only modestly (see column 3 in Tables 2 and 3 for some of the model sizes considered in this article). As a result, traditional methods for making a design robust to an entire model space space–for instance, Li and Nachtsheim (2000), Loeppky et al. (2007), Jones et al. (2009), and Smucker et al. (2012)–are computationally infeasible. Smucker and Drew (2015) present a method that uses a small subset $\mathcal{S}_1$ of the entire model space $\mathcal{F}$ to construct a design. They found that this "approximate model space" makes designs reasonably robust to the entire model space. We use this idea for the AMT model space. Note that we choose $\mathcal{S}_1$ randomly instead of systematically; the designs obtained using randomly chosen approximate model spaces in Smucker and Drew (2015) were of about the same quality as those using systematically chosen models. A more general methodology of systematic selection of approximate model spaces is of interest as future work in this area.

Thus, to construct model-robust designs when the number of models m in $\mathcal{F}$ is large, we do the following:

1. Choose a set of $s_1 << m$ models at random from $\mathcal{F}$. Call this subset $\mathcal{S}_1$.
2. Construct a design robust to $\mathcal{S}_1$ using the algorithm in Section and the criteria in Section .
3. Use the model-robust criterion to evaluate the design with respect to $\mathcal{F}$ or, if $\mathcal{F}$ is too large, with respect to $\mathcal{S}_2$, another randomly chosen subset with s_2 elements.

Based on results in Smucker and Drew (2015), we used s_1's ranging from 16 to 128 and s_2 of 10,000 in this article. In the subsequent sections, we refer to our proposed model-robust designs as AMT designs.

Empirical results

In this section, we demonstrate for several examples the performance of the AMT designs constructed by our algorithm, and compare them when possible to the Bayesian-optimal designs of DuMouchel and Jones (1994) as implemented in JMP.

Preliminaries

The examples considered in the course of this work, including those reported in this paper, are listed in Table 1. In the tables of results in this section, examples are referenced by their number in Table 1.

Table 1: Examples considered. Constraint types L/U denote component lower and upper bounds; type M denotes multicomponent constraints. Note: examples 5 and 6 are omitted from this table because they are not included in the article. Example 5 is a six-component example from Smith (2005) and example 6 is an 11-component example from Piepel (1982); their information is included in the supplementary material.

Example number	Source	Number of components	Constraint types
1	Goldfarb et al. (2004)	3	L/U
2	Cornell (2002)	4	L/U/M
3	Heredia-Langner et al. (2004)	5	L/U/M
4	Piepel (1983)	6	L/U
7	Fleury et al. (2014)	12	L/U/M
8	Piepel et al. (2005)	21	L/U/M

The AMT model-robust designs were constructed by the algorithm described earlier, using the D and I criteria and the AMT model space. In reporting the quality of the designs, they are evaluated on either the full model space $\mathcal{F}$ or a large random sample from the full space, $\mathcal{S}_2$.

Throughout this section we compare the AMT designs with Bayesian D-optimal designs (DuMouchel and Jones 1994). Bayesian-optimal designs are model-robust in the sense that they provide good estimation of terms specified as "primary", while affording some estimation for terms identified as "potential". There is a Bayesian I-optimal analog, which focuses on good prediction. In our setting, we chose the components to be the primary terms and the higher-order blending terms as potential. The Bayesian designs are constructed giving primary terms an infinite prior variance while assigning potential terms a prior distribution with mean 0 and variance τ^2. With this setup, the number of primary terms must be less than the number of runs, but the sum of the number of primary terms, p_1, and the number of potential terms, p_2, can be greater than the number of runs. For D-optimality, this Bayesian procedure results in choosing the design that optimizes $|X^TX + K/\tau^2|$, where K is $(p_1+p_2)\times(p_1+p_2)$ with 0's on the first p_1 diagonals, 1's on the last p_2 diagonals, and 0's elsewhere. Similarly, for I-optimality the criterion is adjusted to $Tr((X^TX + K/\tau^2)^{-1}M)$. The advantage of the Bayesian approach is its computational simplicity—it is nearly as easy to compute Bayesian optimal designs as it is to compute an optimal design for a specific model. The downsides are that it requires the specification of the tuning parameter τ^2 and the Bayesian criterion itself is not directly interpretable. Of course, the Bayesian designs can be evaluated in terms of D-efficiency and the I-criterion (see the paragraph following), which is what we do in the testing in this section. After some informal experimentation with the tuning parameter, we found that $\tau^2 = 0.001$ seemed to provide reasonable model-robust designs and so we used it for the Bayesian optimal designs in this article. We used JMP (SAS Institute

Inc. 2017) software to construct the Bayesian optimal designs.

We quantify the quality of model-robust designs using D-efficiency and the I-criterion. Conceptually, D-efficiency compares a design with a D-optimal design. For a model f and design ξ, D-efficiency is usually approximated using $E_f(\xi) = \left(\frac{det(M_f(\xi))}{n^{p_f}}\right)^{\frac{1}{p_f}} = \frac{det(M_f(\xi))^{\frac{1}{p_f}}}{n}$, where $M_f(\xi)$ is the information matrix, n is the number of rows in ξ, and p_f is the number of parameters in f. This approximation compares design ξ with a theoretical orthogonal design, and provides a lower bound on the true D-efficiency of ξ. In these highly constrained mixture experiment settings, we expect the designs to be far from orthogonal and thus the D-efficiencies to be very low. For comparing different D-criterion model-robust designs, we use the average D-efficiency among all models f in the model space of interest. We also use the average relative D-efficiency to directly compare two designs. This is simply the ratio of two average D-efficiencies, and is indicated as "relative D-efficiency". For the I-criterion model-robust designs, we quantify their quality by evaluating the Mean Average Prediction Variance (MAPV), where the "average" is over the design space for a particular model f and the "mean" is over the set of models. The ratio of two MAPVs is used to compare two designs directly, and termed Relative MAPV.

Results

We constructed AMT designs for several examples from Table 1 and compared them to Bayesian-optimal designs. AMT-D designs, based on the D-criterion, are given in Table 2, while AMT-I designs, based on the I-criterion, are given in Table 3.

In Table 2, for each experimental scenario, we explored the quality of each AMT design as the number of models m in $\mathcal{S}_1$ was increased, and compared these designs to the Bayesian D-optimal designs described in the previous subsection. The AMT-D designs improve upon the Bayesian D-optimal designs by up to 26%, though in several cases the Bayesian-optimal designs are competitive. Interestingly, there is not a strong, positive relationship between m and the quality of design. This observation leads us to the same recommendation as given in Smucker and Drew (2015), that the experimenter should choose the best design, in terms of the D or I criterion evaluated on $\mathcal{S}_2$, among several designs constructed based on differently-sized $\mathcal{S}_1$. In our evaluation, we have used $\mathcal{S}_1$ ranging in size from 16 to 128. If this strategy is adopted, the AMT designs outperform the Bayesian designs in all cases. If instead just the AMT design with the largest $\mathcal{S}_2$ is used, the superiority of the AMT designs holds but is less striking.

Since I-optimal designs are focused on minimizing the prediction variance of the response, the average variance of prediction (AVP) across the experimental region is commonly used for comparing designs. AVP is simply calculated as the I-criterion value $trace((X'X)^{-1}M)$ divided by the volume of the experimental region.

Table 2: Arbitrary Mixture Terms designs using the D-criterion, denoted as AMT-D-xx, where 'xx' is the size of $\mathcal{S}_1$. Parameters: q is the number of components, n is the number of runs, g_2 is the number of second-order model terms, and g_3 is the number of third-order model terms. All Bayes-D-Optimal designs used $\tau = 0.001$. Average D-efficiency was calculated with respect to the whole model space specified by the parameters. If there are more than 10,000 models in the model space, a random sample of 10,000 models was used to estimate the average D-efficiency for the model space. Relative D-Efficiencies were calculated with respect to the Bayesian D-optimal designs, so that a relative D-efficiency greater than 1 indicates a design was more D-efficient, on average, than the Bayesian D-optimal design. All designs were constructed using 100 algorithm tries.

Example	Parameters	Model space size m	Design	Average D-Efficiency (%)	Relative D-Efficiency
2	$q = 4, n = 10,$ $g_2 = 3, g_3 = 0$	20	AMT-D-10	0.1534	1.06
			AMT-D-16	0.1549	1.07
			AMT-D-20	0.1549	1.07
			Bayes-D	0.1450	1
2	$q = 4, n = 10,$ $g_2 = 3, g_3 = 1$	80	AMT-D-16	5.785e-2	1.01
			AMT-D-32	5.754e-2	1.00
			AMT-D-64	5.780e-2	1.00
			Bayes-D	5.755e-2	1
3	$q = 5, n = 14,$ $g_2 = 4, g_3 = 0$	210	AMT-D-16	2.416e-2	1.10
			AMT-D-32	2.421e-2	1.10
			AMT-D-64	2.417e-2	1.10
			Bayes-D	2.191e-2	1
3	$q = 5, n = 20,$ $g_2 = 4, g_3 = 3$	25200	AMT-D-16	3.087e-3	1.02
			AMT-D-32	3.062e-3	1.02
			AMT-D-64	3.045e-3	1.01
			AMT-D-128	3.062e-3	1.02
			Bayes-D	3.015e-3	1
4	$q = 6, n = 20,$ $g_2 = 7, g_3 = 0$	6435	AMT-D-16	0.1912	1.08
			AMT-D-32	0.1928	1.08
			AMT-D-64	0.1929	1.08
			Bayes-D	0.1778	1
4	$q = 6, n = 30,$ $g_2 = 7, g_3 = 7$	4.98e8	AMT-D-16	1.502e-2	1.24
			AMT-D-32	1.534e-2	1.26
			AMT-D-64	1.478e-2	1.22
			AMT-D-128	1.533e-2	1.26
			Bayes-D	1.215e-2	1
7	$q = 12, n = 40,$ $g_2 = 20, g_3 = 0$	4.07e16	AMT-D-16	9.153e-5	1.03
			AMT-D-32	9.300e-5	1.04
			AMT-D-64	9.102e-5	1.02
			AMT-D-128	9.251e-5	1.04
			Bayes-D	8.902e-5	1

In this case (Table 3), the AMT designs are better than the Bayesian I-optimal design by at least 8% and up to 64%. Note that these AMT-I designs are less stable than the AMT-D design, as demonstrated by the two cases where the "Relative MAPV" column includes ∞. This signals that the design could not estimate all of the models in $\mathcal{S}_2$. While this is not ideal, the designs are still able to estimate a large proportion of the models in the AMT model space, and for those estimable models their prediction capabilities (as quantified by MAPV) are good. These results suggest that for the AMT-I criterion, it is beneficial to try a greater variety of designs with different, and larger, $\mathcal{S}_1$ sizes.

Table 3: Arbitrary Mixture Terms designs using the I-criterion, denoted as AMT-I-xx, where 'xx' is the size of $\mathcal{S}_1$. Parameters: q is the number of components, n is the number of runs, g_2 is the number of second-order model terms, and g_3 is the number of third-order model terms. The number of models r is the size of the approximate model space considered. All Bayes-I-Optimal designs used $\tau = 0.001$. Mean average variance of prediction was calculated with respect to $\mathcal{F}$, unless there were more than 10000 models in the model space. In that case, a random sample of 10000 models ($\mathcal{S}_2$) was used to estimate the mean average variance of prediction for the model space. Relative I-Efficiencies are with respect to the Bayesian I-optimal designs, so that a relative I-efficiency greater than 1 indicates a design more I-efficient, on average, than the Bayesian I-optimal design. For those designs with "Relative MAPV" of ∞, one or more models in $\mathcal{S}_2$ were inestimable, and the proportion estimable is given in parentheses in the "Mean Avg Prediction Var" column. The other number in the "Mean Avg Prediction Var" column is the average prediction variance for all estimable models. All designs were constructed using 100 algorithm tries.

Example	Parameters	Model space size	Design	Mean Avg Prediction Var	Relative MAPV
2	$q = 4, n = 10,$	20	AMT-I-10	0.38	1.08
	$g_2 = 3, g_3 = 0$		AMT-I-16	0.38	1.08
			AMT-I-20	0.38	1.08
			Bayes-I	0.41	1
2	$q = 4, n = 10,$	80	AMT-I-16	0.73	0.83
	$g_2 = 3, g_3 = 1$		AMT-I-32	0.51	1.19
			AMT-I-64	0.48	1.26
			Bayes-I	0.61	1
3	$q = 5, n = 14,$	210	AMT-I-16	0.31	1.10
	$g_2 = 4, g_3 = 0$		AMT-I-32	0.30	1.13
			AMT-I-64	0.30	1.14
			Bayes-I	0.34	1
3	$q = 5, n = 20,$	25200	AMT-I-16	0.28	1.09
	$g_2 = 4, g_3 = 3$		AMT-I-32	0.28	1.10
			AMT-I-64	0.28	1.09
			AMT-I-128	0.29	1.07
			Bayes-I	0.31	1
4	$q = 6, n = 20,$	6435	AMT-I-16	1.38	0.30
	$g_2 = 7, g_3 = 0$		AMT-I-32	0.25	1.64
			AMT-I-64	0.23 (0.99)	∞
			Bayes-I	0.41	1
4	$q = 6, n = 30,$	4.98e8	AMT-I-16	0.32	1.36
	$g_2 = 7, g_3 = 7$		AMT-I-32	0.32 (1.00)	∞
			AMT-I-64	0.29	1.49
			AMT-I-128	0.30	1.47
			Bayes-I	0.43	1

The 21-component example

The proposed method is especially useful for generating designs that account for some higher-order behavior when there are limited resources, and this situation is pushed to its extreme for the 21-component example of Piepel et al. (2005), where a full quadratic model would require at least 231 runs and a special-cubic model is much worse. Here, we demonstrate the model-robust design construction method for this large example, first assuming that no more than 14 of the 210 second order terms are expected to be active. In this case, there are $\binom{210}{14} \approx 2.39e21$ models in the model space. Further, we generated AMT designs that also account for possible third-order terms.

Because the design is relatively large and complicated, numerical difficulties prevented the construction of AMT-I designs for these difficult scenarios. We ~~will~~ discuss these issues briefly in the next section.

Table 4: AMT designs using the D-criterion, based on 20 algorithm tries. Parameters: q is the number of components, n is the number of runs, g_2 is the number of second-order model terms, and g_3 is the number of third-order model terms. All Bayesian D-Optimal designs used $\tau = 0.001$. Average D-efficiency was calculated with respect to a random sample of 10,000 models. Relative D-Efficiencies were calculated with respect to the Bayesian D-optimal designs, so that a relative D-efficiency greater than 1 indicates a design was more D-efficient, on average, than the Bayesian D-optimal design.

Example	Parameters	Model space size m	Design	Average D-Efficiency (%)	Relative D-Efficiency
8	$q = 21, n = 45,$	2.39e21	AMT-D-16	1.869e-3	1.02
	$g_2 = 14, g_3 = 0$		AMT-D-32	1.811e-3	0.99
			AMT-D-64	1.850e-3	1.01
			AMT-D-128	1.821e-3	1.00
			Bayes-D	1.828e-3	1
8	$q = 21, n = 45,$	8.22e34	AMT-D-16	3.851e-4	0.96
	$g_2 = 14, g_3 = 5$		AMT-D-32	4.031e-4	1.01
			AMT-D-64	4.087e-4	1.02
			AMT-D-128	4.060e-4	1.02
			Bayes-D	3.992e-4	1

Limitations and challenges

Though the model-robust procedure we described can effectively provide designs when experimental resources are constrained relative to the potential size of the experiment, there are several limitations and challenges that are worth noting.

First, the model-robust design criteria should not be used if resources exist to construct an optimal design for the full model. That is, the model-robust design will typically be only slightly better than the optimal design when evaluated in model-robust terms, but much less effective at estimating the full model. This conclusion is based upon testing in which the number of mixture design points is greater than the number of terms in the full model. In this case, the D-optimal design estimated the full model typically by orders of magnitude better than the AMT design in terms of the D-criterion, while only being slightly worse in terms

of the model space. For I-optimal designs, the conclusions are similar.

The second important limitation is related to the numerical challenges of the I-criterion. Particularly for large, highly constrained mixture scenarios, our implementation failed to produce meaningful designs for the 12-component and 21-component mixture experiments. It's important to note that in principle our proposed procedure can construct model-robust designs for these larger examples. However, issues with numerical precision prevented construction of these large I-criterion-based designs. If dedicated optimization software, such as CPLEX or Gurobi **was used**, these problems might be surmounted.

Third, as pointed out by a reviewer, model discrimination is a concern for the AMT model spaces in this article. That is, it could be the case that a pair of models are both well-estimated (or both allow good prediction), but cannot be easily distinguished from each other. We performed an investigation of this issue by measuring the subspace angle (SA) between models for the AMT designs in Tables 2, 3, and 4, using the procedure discussed in Jones et al. (2007) and implemented in the supplementary material of Smucker and Drew (2015). This method measures how close the linear spaces spanned by the columns of two model matrices are from each other. If two models are 90 degrees apart, they are completely discriminated; if the measure is 0, they are indistinguishable. For the designs in Tables 2-4, we considered every pair of models for small AMT model spaces, whereas for model spaces with more than 500 models, we chose $\binom{500}{2} = 124{,}750$ pairs at random. Across all the designs, the median SA (among the pairs of models considered for a particular design) was never below 70 degrees and for most of the designs the median was above 80 degrees. On the other hand, for most of the designs, the minimum was less than 10 degrees and many had a minimum of 0. The larger designs tended to have more discriminated models, while the two Example 2 scenarios in Table had first quartile SA's between 30 and 50 degrees. Based on these results, there is little difference between the AMT-D and AMT-I designs in terms of model-discrimination. Overall, model discrimination does not seem to be of great concern for these designs.

Fourth, multicollinearity is unavoidable in highly constrained mixture experiments. This has been noted in the literature (e.g. John 1984; Prescott et al. 2002), and in the examples explored here variance inflation factors (VIFs) are often quite high. **As a reviewer pointed out, "large" VIFs (e.g. as judged by rules of thumb about VIFs greater than 5 or 10) do not necessarily imply numerical instability and may not be of great concern. If there is concern, evident ill-conditioning can be remedied by** ridge regression (John 1984) or alternative models (e.g. Prescott et al. 2002; Cornell and Gorman 2003; Kang et al. 2016).

Finally, a related problem is that for experiments like those considered in this article with (a) severely constrained design spaces; (b) resource constraints that result in models with relatively few error degrees of freedom; and (c) large numbers of models that could be fit, there

is no guarantee that all parameters in all models considered will be estimable. In these cases, VIFs will be large enough (e.g. on the order of $1e20$) to indicate the problem. We suggest that before these designs are implemented, numerical stability and model estimability measures—such as VIF or information matrix condition numbers—be examined for the range of models under consideration. If they reveal inestimable models, the experimenter can either assess whether the proportion of inestimable models is small enough to be acceptable, reconsider the number of assumed model terms, or find more resources for the experiment.

Discussion and conclusion

We have presented a new algorithm which can generate model-robust designs for mixture experiments. It is an adaptation of a heuristic which has been implemented in JMP, and a key feature is that the designs are generated without a candidate list. The model-robust approach that we advocate conceptualizes a set of models, called the Arbitrary Mixture Terms (AMT) space, which requires the experimenter to specify an upper bound on the number of quadratic terms, special-cubic terms, as well as higher-order terms if desired. The key advantage of this model-robust approach is that under reasonable assumptions of sparsity in the higher-order blending terms, designs can be constructed even when there are not enough experimental resources to estimate a mixture model with all second-, third-, and/or higher-order blending terms. As the number of components grows, this issue of resources becomes more and more relevant. In cases with large numbers of components and resource constraints, the experimenter is forced to either artificially reduce the number of components considered or design for a first-order order model even though higher-order terms are plausibly important. The methodology developed here, as well as the Bayesian-optimal designs of DuMouchel and Jones (1994), provide principled alternatives. In this article, we demonstrated that our AMT approach generally outperforms a reasonable implementation of the Bayesian-optimal designs, in terms of the average D- and I-criteria for the AMT model space. On the other hand, when experimental resources are adequate to estimate a full second- or third-order mixture model—as will typically be the case when the number of components are small—our empirical work suggests that practitioners should use the design optimal for the full model rather than AMT designs.

Conceptually, the method we present is quite flexible. For instance, while we have focused on D- and I-optimality, the algorithm could be adapted to any other reasonable criterion of interest. Furthermore, it is not limited to quadratic or special-cubic Scheffé models: Full-cubic Scheffé models, which include terms of the type $x_i x_j(x_i - x_j)$, could be accommodated, as well as Becker models (Becker 1968) (one version of which includes terms such as $\min(x_1, x_2)$). The partial quadratic mixture models discussed in Piepel et al.

(2002) and Smith (2005) would also fit neatly into this framework.

Acknowledgements

The authors express their thanks to Chris Gotwalt, who provided them with details and feedback on JMP's algorithm to construct optimal mixture experiments, and also to the two reviewers who provided many comments and inspired additional considerations. Also, gratitude is expressed to JMP and JMP's Technical Support team—in particular Sarah Seligman—who pursued a solution that allowed the Bayesian D-optimal designs for the 21-component example to be constructed. Finally, we greatly appreciate Jens Mueller of Miami University's Research Computing Support, and his assistance with the Redhawk cluster.

References

(2013), *MATLAB version 8.1.0 (R2013a) and Optimization Toolbox*, The Mathworks, Inc., Natick, Massachusetts.

Bazaraa, M. S., Sherali, H., and Shetty, C. (2006), *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, Inc.

Becker, N. (1968), "Models for the response of a mixture," *Journal of the Royal Statistical Society: Series B (Methodological)*, 30, 349–358.

Borkowski, J. and Piepel, G. (2009), "Uniform Designs for Highly Constrained Mixture Experiments," *Journal of Quality Technology*, 41.

Cook, R. D. and Nachtsheim, C. J. (1980), "A comparison of algorithms for constructing exact D-optimal designs," *Technometrics*, 22, 315–324.

— (1982), "An Analysis of the k-exchange Algorithm," Tech. rep., University of Minnesota, Dept. of Operations and Management Science.

Cornell, J. A. (2002), *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data*, New York, NY: Wiley, 3rd ed.

Cornell, J. A. and Gorman, J. W. (2003), "Two new mixture models: living with collinearity but removing its influence," *Journal of Quality Technology*, 35, 78–88.

CPLEX, I. I. (2016), "V12.7: CPLEX User's Manual," .

Crosier, R. (1986), "The Geometry of Constrained Mixture Experiments," *Technometrics*, 28, 95–102.

Dette, H. (1990), "A generalization of $\mathcal{D}$- and $\mathcal{D}_1$-optimal design in polynomial regression," *Annals of Statistics*, 18, 1784–1804.

Dette, H. and Franke, T. (2001), "Robust Designs for Polynomial Regression by Maximizing a Minimum of $\mathcal{D}$- and $\mathcal{D}_1$-Efficiencies," *The Annals of Statistics*, 29(4), 1024–1049.

DuMouchel, W. and Jones, B. (1994), "A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model," *Technometrics*, 36(1), 37–47.

Fang, K.-T. and Yang, Z.-H. (2000), "On Uniform Design of Experiments with Restricted Mixtures and Generation of Uniform Distribution on Some Domains," *Statistics and Probability Letters*, 46, 113–120.

Fang, Z. and Wiens, D. P. (2000), "Integer-Valued, Minimax Robust Designs for Estimation and Extrapolation in Heteroscedastic, Approximate Linear Models," *Journal of the American Statistical Association*, 95(451), 807–818.

Fedorov, V. V. (1972), *Theory of optimal experiments*, Elsevier.

Fleury, B., Godon, N., Ayral, A., Perret, D., Dussossoy, J., and Gin, S. (2014), "Development of an experimental design to investigate the effects of R7T7 glass composition on the residual rate of alteration," *Procedia Materials Science*, 7, 193–201.

Goldfarb, H., Anderson-Cook, C., Borror, C., and Montgomery, D. (2004), "Fraction of Design Space Plots for Assesing Mixture and Mixture-Process Designs," *Journal of Quality Technology*, 36, 169–179.

Goos, P. and Jones, B. (2011), *Optimal Design of Experiments: A Case Study Approach*, Wiley.

Gotwalt, C. and Lancaster, L. (personal communication).

Gurobi Optimization, I. (2016), "Gurobi Optimizer Reference Manual," .

Gutiérrez, I. G.-C. and Martın-Martın, R. (2016), "Optimal Robust Design for Mixture Experiments," Tech. rep.

Heredia-Langner, A., Montgomery, D. C., Carlyle, W. M., and Borror, C. M. (2004), "Model-Robust Optimal Designs: A Genetic Algorithm Approach," *Journal of Quality Technology*, 36(3), 263–279.

John, R. C. S. (1984), "Experiments with mixtures, ill-conditioning, and ridge regression," *Journal of Quality Technology*, 16, 81–96.

Jones, B., Li, W., Nachtsheim, C. J., and Ye, K. Q. (2007), "Model Discrimination—Another Perspective of Model Robustness," *Journal of Statistical Planning and Inference*, 137(5), 1576–1583.

Jones, B. A., Li, W., Nachtsheim, C. J., and Ye, K. Q. (2009), "Model-robust supersaturated and partially supersaturated designs," *Journal of Statistical Planning and Inference*, 139, 45–53.

Kang, L., Salgado, J. C., and Brenneman, W. A. (2016), "Comparing the Slack-Variable Mixture Model With Other Alternatives," *Technometrics*, 58, 255–268.

Läuter, E. (1974), "Experimental design in a class of models," *Mathematische Operationsforschung und Statistik*, 5, 379–398.

Li, W. and Nachtsheim, C. J. (2000), "Model-Robust Factorial Designs," *Technometrics*, 42(4), 345–352.

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

Limmun, W., Chomtee, B., and Borkowski, J. J. (2018), "The Construction of a Model-Robust IV-Optimal Mixture Designs Using a Genetic Algorithm," *Mathematical and Computational Applications*, 23, 25.

Loeppky, J. L., Sitter, R. R., and Tang, B. (2007), "Nonregular Designs with Desirable Projection Properties," *Technometrics*, 49(4), 454–467.

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

Ockuly, R. A., Weese, M. L., Smucker, B. J., Edwards, D. J., and Chang, L. (2017), "Response Surface Experiments: A Meta-Analysis," 164, 64–75.

Petersen, K. B. and Pedersen, M. S. (2012), "The Matrix Cookbook," Version 20121115.

Piepel, G., Anderson, C., and Redgate, P. (1993), "Response surface designs for irregularly-shaped regions (Parts 1, 2, and 3)," in *1993 Proceedings of the Section on Physical and Engineering Sciences*, American Statistical Association Alexandria, VA, pp. 205–227.

Piepel, G. F. (1982), "Component Effects in Constrained Mixture Experiments," *Technometrics*, 24, 29–39.

— (1983), "Calculating Centroids in Constrained Mixture Experiments," *Technometrics*, 25, 279–283.

Piepel, G. F., Cooley, S. K., and Jones, B. (2005), "Construction of a 21-Component Layered Mixture Experiment Design Using a New Mixture Coordinate-Exchange Algorithm," *Quality Engineering*, 17, 579–594.

Piepel, G. F., Szychowski, J. M., and Loeppky, J. L. (2002), "Augmenting Scheffé linear mixture models with squared and/or crossproduct terms," *Journal of Quality Technology*, 34, 297–314.

Prescott, P., Dean, A. M., Draper, N. R., and Lewis, S. M. (2002), "Mixture experiments: ILL-conditioning and quadratic model specification," *Technometrics*, 44, 260–268.

SAS Institute Inc. (2017), "JMP® 13 Design of Experiments Guide, Second Edition," .

Scheffé, H. (1958), "Experiments with mixtures," *Journal of the Royal Statistical Society: Series B (Methodological)*, 20, 344–360.

Smith, W. (2005), *Experimental Design for Formulation*, SIAM.

Smucker, B. J., del Castillo, E., and Rosenberger, J. L. (2011), "Exchange Algorithms for Constructing Model-Robust Experimental Designs," *Journal of Quality Technology*, 43(1), 28–42.

— (2012), "Model-Robust Two-Level Designs Using Coordinate Exchange Algorithms and a Maximin Criterion," *Technometrics*, 54(4), 367–275.

Smucker, B. J. and Drew, N. M. (2015), "Approximate Model Spaces for Model-Robust Experiment Design," *Technometrics*, 57, 54–63.

Tsai, P.-W. and Gilmour, S. G. (2010), "A General Criterion for Factorial Designs under Model Uncertainty," *Technometrics*, 52(2), 231–242.

Welch, W. J. (1983), "A Mean Squared Error Criterion for the Design of Experiments," *Biometrika*, 70(1), 205–213.

A Appendix: derivatives for the D- and I-optimal objective functions

The derivative of the objective function was given to *fmincon* in the algorithm. The derivative for the D-optimal objective function was derived as follows.

$$\begin{aligned}
\frac{\partial d_D(\xi)}{\partial x_{ij}} &= -Tr[(X^TX)^{-1}\frac{\partial X^TX}{\partial x_{ij}}] \quad \text{(Petersen and Pedersen 2012, eq. (43))} \\
&= -Tr[(X^TX)^{-1}\frac{\partial}{\partial x_{ij}}(f(x_i)f^T(x_i))] \\
&= -2Tr[(X^TX)^{-1}(\frac{\partial f(x_i)}{\partial x_{ij}})f^T(x_i)] \quad \text{(Petersen and Pedersen 2012, eq. (93))} \\
&= -2Tr[f^T(x_i)(X^TX)^{-1}(\frac{\partial f(x_i)}{\partial x_{ij}})] \\
&= -2f^T(x_i)(X^TX)^{-1}(\frac{\partial f(x_i)}{\partial x_{ij}})
\end{aligned}$$

The derivative for the I-optimal objective function was derived as follows.

$$
\begin{aligned}
\frac{\partial d_I(\xi)}{\partial x_{ij}} &= Tr[(\frac{\partial Tr((X^TX)^{-1}M)}{\partial(X^TX)})^T \frac{\partial(X^TX)}{\partial x_{ij}}] \quad \text{(Petersen and Pedersen 2012, eq. (137))} \\
&= Tr[-(X^TX)^{-1}M(X^TX)^{-1}\frac{\partial(X^TX)}{\partial x_{ij}}] \quad \text{(Petersen and Pedersen 2012, eq. (124))} \\
&= -Tr[(X^TX)^{-1}M(X^TX)^{-1}(2\frac{\partial f(x_i)}{\partial x_{ij}}f^T(x_i))] \quad \text{(Petersen and Pedersen 2012, eq. (93))} \\
&= -2Tr[f^T(x_i)(X^TX)^{-1}M(X^TX)^{-1}(\frac{\partial f(x_i)}{\partial x_{ij}})] \\
&= -2f^T(x_i)(X^TX)^{-1}M(X^TX)^{-1}(\frac{\partial f(x_i)}{\partial x_{ij}})
\end{aligned}
$$

Supplementary material

Supplementary material for this article is included online and consists of the following:

- **Code** Matlab code that can be used to construct model-robust designs based on the I- or D-criterion.
- **Designs** Designs evaluated in Tables 2, 3, and 4.
- **Design_evaluation** Code for the evaluation of the designs in Tables 2, 3, and 4.