DEPARTMENT OF ENVIRONMENT,
TECHNOLOGY AND TECHNOLOGY MANAGEMENT

Prespecified Factor-Level Combinations in the Optimal Design of Mixture-Process Variable Experiments

Bart De Ketelaere, Peter Goos & Kristof Brijs

UNIVERSITY OF ANTWERP
Faculty of Applied Economics

EARNED EXCELLENCE
AACSB ACCREDITED
EST. 1916
THE BEST BUSINESS SCHOOLS IN THE WORLD

Stadscampus
Prinsstraat 13, B.226
BE-2000 Antwerpen
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
http://www.ua.ac.be/tew

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENVIRONMENT,
TECHNOLOGY AND TECHNOLOGY MANAGEMENT

Prespecified Factor-Level Combinations in the Optimal Design of Mixture-Process Variable Experiments

Bart De Ketelaere, Peter Goos & Kristof Brijs

RESEARCH PAPER 2011-001
FEBRUARY 2011

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@ua.ac.be

D/2011/1169/001

Prespecified Factor-Level Combinations in the Optimal Design of Mixture-Process Variable Experiments

Bart De Ketelaere

Katholieke Universiteit Leuven

Peter Goos

Universiteit Antwerpen
Erasmus Universiteit Rotterdam

Kristof Brijs

Katholieke Universiteit Leuven

Abstract

Designed experiments allow researchers to explore and understand the relation between sets of experimental factors and one or more responses. However, there are many cases where the experimenter cannot determine the factor levels freely. In this article, we study a baking experiment where the interest is in understanding the relationship between the characteristics of flour and the bread quality. The experimental factors are six mixture components and one observable process variable, so that the baking experiment is a mixture-process variable (MPV) experiment. The starting point for the study is an available stock of 30 flour samples for which the mixture component proportions and the level of the process variable have been determined using an expensive chemical analysis. The challenge for the experimenter is to determine which flour samples, or which combinations of flour samples, should be used in the actual experiment. Since the design region is highly constrained and the usable factor levels are dictated by the set of 30 initial flour samples, classical MPV designs are not applicable and an optimal design approach is required. We compare two different design strategies, one based on a point-exchange algorithm and another based on a coordinate-exchange algorithm. The coordinate-exchange algorithm is the most efficient one since it is not bounded by a given candidate set of points. However, it requires the calculation of the convex hull of the set of 30 flour samples along with the related constraints. Because of the high dimensionality of the problem, determining the convex hull and using it to create the constraints for the experimental design is a tedious process. In contrast, the point-exchange algorithm starts from a candidate set of possible factor level combinations, a limitation that lowers the efficiency of the design but results in a design that is very easy to implement.

Keywords: coordinate-exchange algorithm, D-optimality, design augmentation, follow-up design, mixture-process variable experiment, point-exchange algorithm.

1 Introduction

Designed experiments are a very powerful tool to quantify the effect of the factors on the response in production processes, or, ultimately, to determine which factor level combinations provide optimal output quality. A key feature of designed experiments is that the researcher varies the factor levels systematically, according to a certain experimental plan. In the food industry however, there are cases in which the experimenter has little or no flexibility in the factor levels that can be used. In this paper, we study a baking experiment in which the factor levels are to a large extent observational in nature, and, hence, cannot be set freely by the experimenter.

In the baking experiment, the impact of seven factors on the quality of a specific type of bread was investigated. Six of the factors, labeled x_1 to x_6, were the proportions of the six chemical components of the flour. Therefore, the levels of these factors sum to one. The seventh factor, which is independent from the other six and which we denote by z, is a continuous one which is related to another quality characteristic of the flour. For a given flour, it is very expensive and time consuming to determine its exact characteristics, as measured by the factors x_1 to x_6 and z. The available resources allowed for the chemical analysis of 30 samples only. Another limitation of the size of the study was that the baking process was elaborate. As a result, it was desired to keep the number of baking experiments to a strict minimum. The experimental strategy of the researchers was therefore to buy a stock of 30 flour samples from various suppliers from all over Europe, and use these samples in a two-stage experiment. Each of the 30 flour samples underwent a chemical analysis to determine the corresponding seven factor levels. The observational factor levels obtained in this way are shown in Table 1. A two-stage experiment was used because the researchers were unsure whether the 30 samples were sufficiently different to give sufficient variation in the responses to fit a statistical model. A small initial experiment was conducted first. Because this was successful, a larger follow-up experiment was then designed.

When planning the baking experiment, the researchers did not only consider the use of the 30 pure flour samples, but they were willing to mix these samples to create “new” flours to bake bread from. Blending the 30 original flour samples would not necessitate additional chemical analyses, but it would drastically increase the set of candidate flours to use in the experiment and possibly allow a better modeling of the relationship between the seven factors and the response. The most obvious new samples to generate were binary and ternary blends of the original samples. These blends had equal proportions for each of the blended samples (i.e., 50% for the two samples in each binary blend, and 33% for the three samples involved in a ternary blend). The baking experiment therefore took the form of a mixture experiment involving six classical mixture factors (x_1 to x_6) that sum to one and one additional uncontrollable continuous factor z. Such an extra factor in a mixture experiment is usually referred to as a process variable.

Table 1: Flour composition for 30 samples. The six mixture variables are denoted by x_1-x_6, while the observable but uncontrollable process variable is denoted by z.

Sample	x_1	x_2	x_3	x_4	x_5	x_6	z
1	0.074	0.042	0.679	0.015	0.032	0.158	11.92
2	0.071	0.040	0.694	0.014	0.037	0.145	8.04
3	0.061	0.028	0.712	0.012	0.031	0.156	6.06
4	0.068	0.032	0.709	0.013	0.032	0.146	19.44
5	0.062	0.037	0.683	0.014	0.040	0.164	4.36
6	0.075	0.033	0.654	0.014	0.027	0.197	4.64
7	0.073	0.028	0.641	0.017	0.036	0.205	8.56
8	0.064	0.027	0.644	0.015	0.035	0.216	3.44
9	0.072	0.042	0.686	0.015	0.033	0.152	3.94
10	0.075	0.053	0.630	0.016	0.036	0.190	7.32
11	0.065	0.043	0.682	0.015	0.030	0.166	7.34
12	0.071	0.045	0.680	0.014	0.029	0.161	13.02
13	0.083	0.042	0.668	0.016	0.029	0.161	3.98
14	0.063	0.043	0.680	0.014	0.031	0.170	11.74
15	0.073	0.051	0.666	0.014	0.036	0.162	10.80
16	0.072	0.052	0.660	0.017	0.032	0.168	10.34
17	0.088	0.051	0.698	0.016	0.035	0.112	5.08
18	0.075	0.044	0.715	0.014	0.034	0.119	15.54
19	0.072	0.046	0.667	0.015	0.045	0.156	3.82
20	0.072	0.052	0.704	0.017	0.039	0.117	9.60
21	0.076	0.045	0.736	0.014	0.030	0.099	19.76
22	0.067	0.045	0.688	0.014	0.036	0.150	7.16
23	0.070	0.036	0.703	0.014	0.036	0.141	4.20
24	0.076	0.048	0.671	0.015	0.043	0.148	5.24
25	0.070	0.052	0.675	0.017	0.042	0.146	4.34
26	0.058	0.052	0.654	0.014	0.037	0.185	10.14
27	0.087	0.048	0.661	0.016	0.036	0.152	6.28
28	0.070	0.055	0.662	0.014	0.038	0.161	3.72
29	0.068	0.052	0.685	0.016	0.037	0.142	2.68
30	0.082	0.047	0.696	0.014	0.039	0.123	3.92

Summarizing, the general characteristics of the baking experiment are (i) an expensive measurement process that limits the number of samples for which factor levels can be determined, (ii) an expensive measurement process for assessing the output quality of the baking process, (iii) the inability to control the seven factor levels of a stock of samples to start from, (iv) the ability to mix samples from this stock to obtain new samples that can be used for processing and model building purposes and (v) a mixture type of experiment with an extra uncontrollable process variable.

The literature dealing with optimal experimental designs for applications with pre-specified combinations of factor levels is limited. Nachtsheim (1989) describes the problem of constructing D-optimal designs with covariates that are fixed and specified in advance. Sexton et al. (2006) describe an example where part of the experiment is designed, and part is observational. Anderson-Cook and Robinson (2009) discuss the construction of optimal designs for a screening study for a production process involving five input factors whose levels had been previously established. They summarize the most important drawbacks of such pre-specified observational setting when compared to a designed experiment. First, establishing causation is not possible, and the result is restricted to understanding the empirical relationship between the response and changes in the factor settings. Second, correlation among factors cannot be eliminated meaning that the least squares estimates of the regression model parameters will be dependent and less precise than from a designed experiment. Based on these observations, Anderson-Cook and Robinson (2009) clearly advocate the use of designed experiments, with levels that are set by the experimenter, whenever possible.

The theory of mixture designs and corresponding analysis techniques are well described in the literature but are generally more complicated than classical techniques handling regular designs. This is mainly due to the constraints that are imposed on the mixture: the sum of all components is a constant, and all proportions lie between zero and the individual components' sum (Box, Hunter and Hunter, 1978). In mixture experiments in the food industry, there are often also additional constraints on the individual component proportions, as well as on linear combinations of the components (see, e.g., Dingstad et al., 2003; Rehman et al. 2007). This complicates the experimental design because experiments involving multicomponent constraints often have an irregularly shaped experimental region, which necessitates the use of computer aided design of experiments (Cornell, 2002). Another complication is the presence of a process variable in the experiment. The design and analysis of mixture experiments with process variables is discussed in Naes et al. (1998), Cornell (2002), Mage and Naes (2005) and Goos and Donev (2007).

In this paper, we describe the search for an ideal set of pure samples, binary blends and/or ternary blends to be used in the baking experiment, taking into account the presence of the process variable and starting from a set of observational factor level combinations. The selected set of samples and blends should allow for a precise estimation of the relationship between the factors and the response. The next section of the paper provides a discussion of various alternative models that can be used for data from mixture-process variable experiments. In Section 3, we explain how mixture-process variable experiments can be designed. We pay special attention to optimal experimental design because that is the approach that we used for designing the baking experiment. In Section 4, we discuss how we constructed various alternative designs for the baking experiment. In Section 5, we quantify the loss of information contained within our experimental design due to the facts that the factor levels in the application were prespecified and only binary and ternary blends were considered. To that end, we compared the results from a point-exchange algorithm to find optimal experimental designs (which is appropriate for situations in which factor levels are pre-specified and a limited number of factor level combinations are used) with those from a coordinate-exchange algorithm (which assumes that the factor levels are not fixed in advance and any combination of the factor levels is possible). We conclude the article with a discussion and a summary of the most important findings from our study and with a few suggestions for future research.

2 Mixture-Process Variable Model

2.1 General model

Models for analyzing data from mixture-process variable experiments are usually obtained by combining traditional Scheffé type models for the mixture variables with response surface models for the process variables. For example, a common mixture-process variable model is obtained by crossing the second-order Scheffé model,

$$E(Y) = \sum_{k=1}^{q} \beta_k x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \beta_{kl} x_k x_l,$$

where q is the number of mixture components and x_k is the proportion of component k in the mixture, with a main-effects-plus-two-factor-interactions model in the process variables,

$$E(Y) = \alpha_0 + \sum_{k=1}^{m} \alpha_i z_i + \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} \alpha_{ij} z_i z_j,$$

where m is the number of process variables and z_i represents the setting of the ith process variable. The combined model can be written as

$$\begin{aligned} Y = {} & \sum_{k=1}^{q} \gamma_k^0 x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^0 x_k x_l \\ & + \sum_{i=1}^{m} \left[\sum_{k=1}^{q} \gamma_k^i x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^i x_k x_l \right] z_i \\ & + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left[\sum_{k=1}^{q} \gamma_k^{ij} x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^{ij} x_k x_l \right] z_i z_j + \varepsilon. \end{aligned} \tag{1}$$

In this expression, the terms

$$\sum_{k=1}^{q} \gamma_k^0 x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^0 x_k x_l$$

correspond to the linear and non-linear blending properties of the mixture components. Each term

$$\left[\sum_{k=1}^{q} \gamma_k^i x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^i x_k x_l \right] z_i$$

contains the linear effect of the ith process variable z_i on the components' blending properties, and terms of the form

$$\left[\sum_{k=1}^{q} \gamma_k^{ij} x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^{ij} x_k x_l \right] z_i z_j$$

describe the interaction effect of process variables z_i and z_j on the blending properties. Note that neither the Scheffé model nor the mixture-process variable model have an intercept to avoid perfect collinearity of the intercept term with the q linear blending terms.

A major problem with the combined model in Equation (1) is that its number of parameters increases rapidly with the number of mixture components and the number of process variables. For q components and m process variables, the number of parameters p in the model amounts to $[q + q(q-1)] \times [m + m(m-1)]$. The large number of parameters in the model implies that at least $[q + q(q-1)] \times [m + m(m-1)]$ runs are required in the experiment, and inspired Kowalski et al. (2000) to suggest a more parsimonious second-order model. Their suggested model assumes that there are no interaction effects of the process variables on the non-linear blending properties (i.e., $\gamma_{kl}^{ij} = 0$ and $\gamma_k^{ij} = a_{ij}$). Also, it is assumed that there is no linear effect of the process variables on the nonlinear blending properties either (i.e., $\gamma_{kl}^i = 0$). The nonlinear blending properties and the interactions between the process variables therefore enter the model only additively. Another difference between the model proposed by Kowalski et al. (2000) and the one in Equation (1) is that the former also involves quadratic effects in the process variables:

$$\begin{aligned} Y = & \sum_{k=1}^{q} \gamma_k^0 x_k + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \gamma_{kl}^0 x_k x_l \\ & + \sum_{i=1}^{m} \left[\sum_{k=1}^{q} \gamma_k^i x_k \right] z_i + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \alpha_{ij} z_i z_j + \sum_{i=1}^{m} \alpha_i z_i^2 + \varepsilon. \end{aligned} \tag{2}$$

It is obvious that this model contains $q + q(q-1) + qm + m(m-1) + m$ terms, so that this model is more parsimonious in terms of the number of parameters than the crossed model in most practical situations. Along with a detailed strategy for building a suitable model, Prescott (2004) presents yet another mixture process-variable model, involving $q(q+1)(q+2)/6 + mq(m+q+2)/2$ terms. That model is clearly not as parsimonious in terms of the number of parameters as the one in Equation (2).

In our application, the experimental budget allowed only for a very limited number of runs. Therefore, we decided to use the mixture-process variable model in Equation (2). In matrix notation, that model can be written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3}$$

where $\mathbf{X}$ represents the $n \times p$ model matrix containing the settings and model expansions of all experimental factors, i.e. the mixture components $x_1, \ldots, x_q$ and the process variables $z_1, \ldots, z_m$ at each of the n runs of the experiment, $\boldsymbol{\beta}$ is the vector containing the factor effects γ_k^0, γ_{kl}^0, γ_k^i, α_{ij} and α_i, and $\boldsymbol{\varepsilon}$ is an n-dimensional vector containing the random errors. We assume that

$$\boldsymbol{\varepsilon} \sim N(\mathbf{0}_n, \sigma_\varepsilon^2 \mathbf{I}_n).$$

It has been pointed out by Cornell (1988), Kowalski et al. (2002), Mage and Naes (2005), Naes et al. (2007), and Goos and Donev (2007) that mixture-process variable designs are often run as split-plot designs. This is due to the fact that, in many cases, mixture-process variables are often done by first preparing all the required mixtures, and then feeding the batches to the process one by one. For every batch, the levels of the process variables are changed multiple times. In this approach, the levels of the factors related to the mixtures (i.e., the mixture components' proportions) are not reset independently for each experimental run. The resulting design is a split-plot design.

2.2 The model for the case study

In our case, there are six mixture variables, $x_1, \ldots, x_6$, and only one process variable, z. Hence, $q = 6$ and $m = 1$. As a result, our model is given by

$$Y = \sum_{k=1}^{6} \gamma_k^0 x_k + \sum_{k=1}^{5} \sum_{l=k+1}^{6} \gamma_{kl}^0 x_k x_l + \left[\sum_{k=1}^{6} \gamma_k x_k \right] z + \alpha z^2 + \varepsilon \tag{4}$$

and involves $6 + 15 + 6 + 1 = 28$ parameters. As a result, at least 28 runs are required to estimate the mixture-process variable model.

An important practical issue in our case is that the ultimate design is composed of an initial design of seven runs and 24 additional runs obtained using an optimal experimental design approach. The 24 additional runs were done in groups of eight in three consecutive weeks. To capture the random week-to-week variation in the model, we would generally advise the inclusion of a separate block effect in the model for all the runs from the initial design and for the three weeks of subsequent experimentation. The estimation of the model including the four block effects would require four additional experimental runs. In our case, however, the inclusion of the block effects in the model was not required because the company carrying out the mixture-process variable experiment was fully confident that there was no week-to-week variation. To confirm this, they included a reference sample in the initial design as well as in each week for the follow-up experiment. The analysis of the data for the four reference samples was reassuring in the sense that it indicated that there was no substantial week-to-week variation.

The baking experiment did not have a split-plot structure because only one level of the process variable z was used for every combination of the mixture component proportions x_i. This is because the level of z was an observed factor level that could not be set by the researchers.

3 Design of Mixture-Process Variable Experiments

3.1 Different approaches

In general, mixture-process variable experiments can be designed by combining mixture designs with process-variable designs. For unconstrained simplex-shaped mixture design regions, this is done by crossing simplex-lattice or simplex-centroid designs for the mixture components and factorial or fractional factorial arrangements for the process variables (see, for example, Cornell (2002), Section 7.1, and Smith (2005), Section 13.2). This way of constructing a MPV design is simple but often leads to large experimental designs. Smaller designs for unconstrained design regions were proposed by Cornell and Gorman (1984), Czitrom (1988, 1989) and Kowalski et al. (2000, 2002). The fact that MPV experiments are often conducted using a split-plot structure was addressed by Cornell (1988, 2002) and Kowalski et al. (2002). The design of MPV experiments was also discussed in Goos and Donev (2007), who suggest using the flexible optimal design approach as a basis for constructing efficient designs. This optimal design approach is indispensable when the design region is constrained (as in our application) because the classical simplex-lattice of simplex-centroid designs can then no longer be used.

3.2 Optimal design

A primary goal of an experimenter is usually to estimate the parameter vector $\boldsymbol{\beta}$ in Equation (3) as precisely as the available budget for the experiment allows. In technical terms, this means that the experimenter desires the ordinary least squares estimator of the parameter vector $\boldsymbol{\beta}$,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y},$$

to have a small covariance matrix

$$\mathrm{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.$$

In the optimal experimental design literature, by far the most commonly used approach to achieve this goal is to seek a design that minimizes the determinant of that covariance matrix. Such a design is called D-optimal, and the corresponding design selection criterion is referred to as the D-optimality criterion. An alternative way to define the D-optimality criterion involves the Fisher information matrix, $\sigma^{-2}(\mathbf{X}'\mathbf{X})$, or its scaled version, $\mathbf{X}'\mathbf{X}$: a D-optimal design is a design that maximizes the determinant of the (scaled) information matrix. The determinant $|\mathbf{X}'\mathbf{X}|$ is the D-criterion value of a design with model matrix $\mathbf{X}$.

In order to compare alternative design options in this paper, we report relative D-efficiencies

$$\left\{\frac{|\mathbf{X}_2'\mathbf{X}_2|^{-1}}{|\mathbf{X}_1'\mathbf{X}_1|^{-1}}\right\}^{1/p} = \left\{\frac{|\mathbf{X}_1'\mathbf{X}_1|}{|\mathbf{X}_2'\mathbf{X}_2|}\right\}^{1/p}, \tag{5}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ represent the model matrices of the two competing designs and p is the number of parameters in $\boldsymbol{\beta}$. A value larger than one for the D-efficiency implies that the design with model matrix $\mathbf{X}_1$ outperforms the one with model matrix $\mathbf{X}_2$.

In the real-life design problem that we study in this paper, we are faced with a situation in which an initial 7-run design needs to be augmented with 24 new runs. In general, the augmentation of an initial design using the D-optimality criterion requires determining the follow-up design with model matrix $\mathbf{X}_{\text{fu}}$ so that the determinant of

$$\text{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}_{\text{init}}'\mathbf{X}_{\text{init}} + \mathbf{X}_{\text{fu}}'\mathbf{X}_{\text{fu}})^{-1},$$

with $\mathbf{X}_{\text{init}}$ the model matrix for the initial design, is minimal, or, alternatively, the D-criterion value $|\mathbf{X}_{\text{init}}'\mathbf{X}_{\text{init}} + \mathbf{X}_{\text{fu}}'\mathbf{X}_{\text{fu}}|$ is maximal.

3.3 Algorithms

Except in very specific instances, the construction of D-optimal designs requires a computerized search. The algorithms used in such a computerized search can be classified in two categories.

The first kind of algorithm is a point-exchange algorithm. Point-exchange algorithms were popularized by Fedorov (1972), after which several adaptations to the Fedorov algorithm have been proposed (e.g., Johnson and Nachtsheim, 1983; Cook and Nachtsheim, 1989; Atkinson and Donev, 1989). A key feature of point-exchange algorithms is that they require a user-specified candidate set as input. Roughly speaking, this candidate set is the set of all possible combinations of factor levels. Every possible factor level combination is typically called a candidate point. The point-exchange algorithms use this set of candidate points to generate a starting design (at least part of that design is a randomly selected subset of the candidate set), which they then try to improve by replacing factor level combinations of the current design by candidate points. The algorithms continue exchanging factor level combinations until the determinant of the covariance matrix $\text{cov}(\hat{\boldsymbol{\beta}})$ or the information matrix can no longer be improved. The generation of a starting design and the subsequent point-wise improvement procedure are repeated a pre-specified number of times to increase the likelihood that the best design in terms of the D-optimality criterion is found.

The second kind of algorithm, the coordinate-exchange algorithm, was introduced in the literature by Meyer and Nachtsheim (1995) and offers the advantage that it does not require the explicit specification of a candidate set. The starting design for this algorithm is determined completely at random. Then, the coordinate-exchange algorithm attempts to improve the level of each factor at each run of the starting design, one by one. If a better level is found for a level of one of the factors in a run, then that level is changed to the better value. This coordinate-wise improvement procedure is continued until none of the individual levels of variables can be improved any more. As in the point-exchange algorithm, the generation of the starting design and the improvement procedure are repeated a pre-specified number of times.

The original coordinate-exchange algorithm assumes that the factor levels can be changed independently. In mixture experiments and mixture-process variable experiments however, the levels of the mixture components cannot be changed independently because the sum of all the mixture components' proportions always has to be one. Piepel et al. (2005) therefore propose a mixture coordinate-exchange algorithm that deals with this problem. The key feature of this algorithm is that whenever the proportion of a certain mixture component is modified, the proportions of all other mixture components are adapted so that their pairwise ratios remain constant.

In general, point-exchange algorithms have two drawbacks. First, their computation time grows exponentially with the number of factors and with the number of levels considered for each of these. Second, in the presence of constraints on the factor levels, it may be difficult to produce a good candidate set. This is because all the combinations in the candidate set have to satisfy the constraints and because it is not known what factor levels may be optimal when the experimental region is constrained. In the baking experiment however, the observational nature of the factor levels of the 30 original samples and the decision to consider only binary and ternary blends of them automatically leads to a finite candidate set and dictates the use of a point-exchange algorithm. In Section 5, we demonstrate that considering all possible blends of the original samples naturally leads to the use of a coordinate-exchange algorithm, with a large set of constraints on the proportions of the mixture components.

4 The baking experiment

Because of the expensive baking process required to measure the response, the researchers decided to start with a small initial designed experiment to examine whether the observational factor levels were sufficiently informative, i.e. to verify whether sufficient spread

Table 2: Flour composition for the initial design of seven samples. The six mixture component proportions are denoted by x_1-x_6, while the observable but uncontrollable process variable is denoted by z.

Sample	x_1	x_2	x_3	x_4	x_5	x_6	z
3	0.061	0.028	0.712	0.012	0.031	0.156	6.06
6	0.075	0.033	0.654	0.014	0.027	0.197	4.64
8	0.064	0.027	0.644	0.015	0.035	0.216	3.44
9	0.072	0.042	0.686	0.015	0.033	0.152	3.94
17	0.088	0.051	0.698	0.016	0.035	0.112	5.08
21	0.076	0.045	0.736	0.014	0.030	0.099	19.76
26	0.058	0.052	0.654	0.014	0.037	0.185	10.14

in the response variable was present. The initial fear of the researchers was that all the baked breads would result in similar values for the response, suggesting that changing the levels of the seven factors would not allow for a better bread quality and that building a useful mixture-process variable model would be impossible. In that case, running a large experiment would be a complete waste of resources. By carrying out a small initial experiment that yielded substantially different values for the response, this concern could be removed. In that case, a larger follow-up experiment would be performed.

The purpose of using the small initial design was thus exploratory, and, therefore, it was created with no particular statistical model in mind. The main goal was to try out a small but diverse set of flours. The initial design ultimately selected consisted of seven of the original 30 flour samples in Table 1. One sample was chosen near the center of the 30-sample dataset, while the six other samples were selected so that their Mahalanobis distance to the center point was large. Using the Mahalanobis distance was more appropriate than using the Euclidean distance because it takes into account the correlation structure of the variables. The initial design is shown in Table 2.

Going through the complete baking process with the initial set of seven flour samples gave a large variability in the response, with some breads having excellent quality, whilst others having a very poor quality. Based on this outcome, the initial design of seven samples was augmented with a follow-up design of 24 new samples. The actual number of additional samples, 24, was chosen as a compromise between the minimal number of samples needed to fit the 28-parameter model in Equation (4), and a practical consideration — a total of eight samples could be baked in a time frame of one week, limiting the time required to conduct the follow-up experiment to three weeks.

The candidate set used for creating the follow-up design that was actually run included the 30 pure samples in Table 1, all 50-50% blends of the 30 original samples (435 candidates) and all 33-33-33% blends (4060 candidates). As a result, in total, there are 4525 candidate points. These candidates were used as input to the modified Fedorov point-exchange algorithm implemented in SAS 9.2. Because point-exchange algorithms often get stuck in local optima and the current design problem involves a large number of candidates, the algorithm was run 5000 times. The commands required to generate the design are listed in Appendix A. The entire search procedure took 300 seconds on a standard laptop computer (Intel Core Duo processor, 2 GHz, 2 GB RAM). The resulting optimal design is shown in Table 3. The first seven points in the optimal design form the initial design. The next 24 runs form the follow-up design. The determinant of the information matrix for the complete design is 9.2665×10^{-138}. The 24 follow-up runs consist of 15 of the original samples in Table 1, eight binary blends, and one ternary blend. The design does not contain any replicates.

As preparing ternary blends is cumbersome, we also studied what information would be lost if we would consider the original samples and binary blends only. To this end, we also computed a D-optimal follow-up design with a restricted candidate set, containing only the pure samples and the 50-50% blends. This restricted candidate set comprises 465 potential runs. With this candidate set, the computation time required to generate the D-optimal follow-up design was about 40 seconds. The resulting design, which is displayed in Table 4, has a determinant of 6.6628×10^{-138}. Hence it has a D-efficiency of 98.83% relative to the design in Table 3. The design involving only pure samples and binary blends thus performs almost equally well as the design constructed using pure samples and binary and ternary blends. The ternary blends are therefore relatively unimportant when compared to the binary blends. This can be verified numerically by computing a D-optimal design starting from a candidate set involving the pure samples and the ternary blends only. This yields a determinant of 9.5933×10^{-139}, so that the design is only 92.22% efficient when compared to the design in Table 3. Thus, a design that does not involve binary blends is substantially less efficient that one that does.

The fact that the smaller candidate set consisting of pure samples and binary blends results in a more efficient design than that based on pure samples and ternary blends should not come as a surprise. The reason is that the model in Equation (4) assumes that second-order blending effects are present. It is well-known from the literature on the optimal design of mixture experiments that, in that case, the midpoints of the edges of the experimental region are highly informative. Obviously, the midpoints of the edges correspond to binary blends consisting of 50% of one sample and 50% of another.

A final interesting exercise is to investigate what the statistical cost is of using an initial 7-run design selected using the Mahalanobis distance and augmenting it with a 24-run follow-up design. A statistically more efficient experimental design is obtained by computing a D-optimal 31-run design using the candidate set involving pure samples and binary and ternary blends. The D-optimal 31-run design we obtained in this fashion is displayed in Table 5. It has a D-criterion value of 1.1661×10^{-137}, and, therefore, a D-efficiency of 1.0082 relative to the design in Table 3 consisting of an initial design and a follow-up

Table 3: Final design for the real-life experiment, with indication of the seven runs of the initial design and the 24 follow-up runs, and the type of blend (pure, binary, ternary). For each run, it is shown which of the 30 original samples have to be mixed.

Type								Type	Samples
of Run	x_1	x_2	x_3	x_4	x_5	x_6	z	of Blend	Used
initial	0.058	0.052	0.654	0.014	0.037	0.185	10.14	pure	26
initial	0.088	0.051	0.698	0.016	0.035	0.112	5.08	pure	17
initial	0.061	0.028	0.712	0.012	0.031	0.156	6.06	pure	3
initial	0.075	0.033	0.654	0.014	0.027	0.197	4.64	pure	6
initial	0.064	0.027	0.644	0.015	0.035	0.216	3.44	pure	8
initial	0.072	0.042	0.686	0.015	0.033	0.152	3.94	pure	9
initial	0.076	0.045	0.736	0.014	0.030	0.099	19.76	pure	21
follow-up	0.061	0.039	0.649	0.014	0.036	0.200	6.79	binary	8, 26
follow-up	0.062	0.037	0.683	0.014	0.040	0.164	4.36	pure	5
follow-up	0.065	0.043	0.682	0.015	0.030	0.166	7.34	pure	11
follow-up	0.065	0.052	0.679	0.015	0.038	0.151	9.87	binary	20, 26
follow-up	0.067	0.028	0.676	0.015	0.034	0.180	7.31	binary	3, 7
follow-up	0.068	0.052	0.685	0.016	0.037	0.142	2.68	pure	29
follow-up	0.068	0.032	0.709	0.013	0.032	0.146	19.44	pure	4
follow-up	0.069	0.037	0.724	0.013	0.031	0.128	12.91	binary	3, 21
follow-up	0.070	0.055	0.662	0.014	0.038	0.161	3.72	pure	28
follow-up	0.070	0.044	0.667	0.017	0.038	0.164	5.19	ternary	7, 25, 29
follow-up	0.071	0.045	0.680	0.014	0.029	0.161	13.02	pure	12
follow-up	0.072	0.046	0.667	0.015	0.045	0.156	3.82	pure	19
follow-up	0.072	0.052	0.704	0.017	0.039	0.117	9.60	pure	20
follow-up	0.072	0.052	0.660	0.017	0.032	0.168	10.34	pure	16
follow-up	0.073	0.028	0.641	0.017	0.036	0.205	8.56	pure	7
follow-up	0.073	0.051	0.666	0.014	0.036	0.162	10.80	pure	15
follow-up	0.075	0.037	0.688	0.015	0.033	0.152	14.16	binary	7, 21
follow-up	0.075	0.053	0.630	0.016	0.036	0.190	7.32	pure	10
follow-up	0.075	0.039	0.702	0.014	0.036	0.134	11.68	binary	4, 30
follow-up	0.078	0.035	0.655	0.017	0.033	0.183	6.27	binary	7, 13
follow-up	0.079	0.040	0.675	0.014	0.033	0.160	4.28	binary	6, 30
follow-up	0.082	0.047	0.696	0.014	0.039	0.123	3.92	pure	30
follow-up	0.083	0.042	0.668	0.016	0.029	0.161	3.98	pure	13
follow-up	0.087	0.048	0.661	0.016	0.036	0.152	6.28	pure	27

Table 4: Alternative design considering only original samples and binary blends.

Type of Run	x_1	x_2	x_3	x_4	x_5	x_6	z	Type of Blend	Samples Used
initial	0.058	0.052	0.654	0.014	0.037	0.185	10.14	pure	26
initial	0.088	0.051	0.698	0.016	0.035	0.112	5.08	pure	17
initial	0.061	0.028	0.712	0.012	0.031	0.156	6.06	pure	3
initial	0.075	0.033	0.654	0.014	0.027	0.197	4.64	pure	6
initial	0.064	0.027	0.644	0.015	0.035	0.216	3.44	pure	8
initial	0.072	0.042	0.686	0.015	0.033	0.152	3.94	pure	9
initial	0.076	0.045	0.736	0.014	0.030	0.099	19.76	pure	21
follow-up	0.061	0.039	0.649	0.014	0.036	0.200	6.79	binary	8, 26
follow-up	0.062	0.037	0.683	0.014	0.040	0.164	4.36	pure	5
follow-up	0.065	0.043	0.682	0.015	0.030	0.166	7.34	pure	11
follow-up	0.065	0.052	0.679	0.015	0.038	0.151	9.87	binary	20, 26
follow-up	0.067	0.028	0.676	0.015	0.034	0.180	7.31	binary	3, 7
follow-up	0.068	0.052	0.685	0.016	0.037	0.142	2.68	pure	29
follow-up	0.068	0.032	0.709	0.013	0.032	0.146	19.44	pure	4
follow-up	0.069	0.037	0.724	0.013	0.031	0.128	12.91	binary	3, 21
follow-up	0.070	0.055	0.662	0.014	0.038	0.161	3.72	pure	28
follow-up	0.071	0.045	0.680	0.014	0.029	0.161	13.02	pure	12
follow-up	0.072	0.040	0.658	0.017	0.039	0.176	6.45	binary	7, 25
follow-up	0.072	0.046	0.667	0.015	0.045	0.156	3.82	pure	19
follow-up	0.072	0.052	0.704	0.017	0.039	0.117	9.60	pure	20
follow-up	0.072	0.052	0.660	0.017	0.032	0.168	10.34	pure	16
follow-up	0.073	0.028	0.641	0.017	0.036	0.205	8.56	pure	7
follow-up	0.073	0.051	0.666	0.014	0.036	0.162	10.80	pure	15
follow-up	0.075	0.037	0.688	0.015	0.033	0.152	14.16	binary	7, 21
follow-up	0.075	0.053	0.630	0.016	0.036	0.190	7.32	pure	10
follow-up	0.075	0.039	0.702	0.014	0.036	0.134	11.68	binary	4, 30
follow-up	0.078	0.035	0.655	0.017	0.033	0.183	6.27	binary	7, 13
follow-up	0.079	0.040	0.675	0.014	0.033	0.160	4.28	binary	6, 30
follow-up	0.082	0.047	0.696	0.014	0.039	0.123	3.92	pure	30
follow-up	0.083	0.042	0.668	0.016	0.029	0.161	3.98	pure	13
follow-up	0.087	0.048	0.661	0.016	0.036	0.152	6.28	pure	27

design. This implies that conducting the experiment in two stages and creating the initial design using the Mahalanobis distance did not lead to a significant loss of efficiency. It can even be verified that six of the seven points selected using the Mahalanobis distance appear in the D-optimal 31-run design. These points are indicated in Table 5 with an asterisk. The only sample from the initial design in Table 2 that does not appear in the D-optimal 31-run design in Table 5 is sample 9.

5 Comparison to Coordinate-Exchange Algorithm

It is obvious that the practical restriction to consider only pure samples, binary blends and ternary blends has a statistical cost. As a matter of fact, it is theoretically possible that blends of more than three samples could be more informative. Also, and more likely, it is possible that considering other proportions than 50-50 and 33-33-33 in the binary and ternary blends, respectively, yields better designs. We therefore also report the D-optimal 24-run follow-up design and the D-optimal 31-run design obtained by assuming that any number of samples can be mixed and that any blending proportion can be used for each of the samples. These designs cannot be found using a point-exchange algorithm because the size of the candidate set for this design problem would be too large. Instead, the designs must be computed using a coordinate-exchange algorithm, which does not require a candidate set. We also relax the practical constraint that the process variable cannot be set independently.

The main difficulty when using the coordinate-exchange algorithm is that it must be possible to create each of the blends in the experiment by combining the 30 original samples. For this to be the case, all the points of the design must lie within the convex hull of the 30 original samples. To ensure that this is the case, we have to determine which of the 30 samples are vertices of the convex hull and use these vertices to create expressions for the hyperplanes or facets of the convex hull. These hyperplanes define the constraints that have to be satisfied for the design points to lie within the convex hull. It turns out that the convex hull of the 30 samples in Table 1 has 234 facets, and, hence, 234 constraints have to be taken into account when generating a D-optimal design with the coordinate-exchange algorithm. In Appendix B, we show how the convex hull of the 30 original samples and the corresponding 234 constraints for the D-optimal design can be determined using MATLAB. In Appendix C, we show how the large number of constraints for our case study can be read into JMP 8 so that a D-optimal 31-run design can be computed using its coordinate-exchange algorithm for mixture experiments. For generating a 24-run follow-up design, we used the “Augment Design” option in JMP 8, which also uses the coordinate-exchange algorithm and allows the uploading of the constraints defined by the convex hull. For each of the two design problems, we constrained the value of the process variable z to lie between 2.68 and 19.76, the minimum and maximum of that variable in the original samples in Table 1. We ran the coordinate-exchange algorithm ten times only. This took us about 25 minutes on a standard laptop (Intel

Table 5: D-optimal 31-run design generated using the candidate set involving pure samples and binary and ternary blends.

Run	x_1	x_2	x_3	x_4	x_5	x_6	z
1^*	0.058	0.052	0.654	0.014	0.037	0.185	10.14
2^*	0.061	0.028	0.712	0.012	0.031	0.156	6.06
3	0.061	0.039	0.649	0.014	0.036	0.200	6.79
4	0.062	0.037	0.683	0.014	0.040	0.164	4.36
5^*	0.064	0.027	0.644	0.015	0.035	0.216	3.44
6	0.065	0.043	0.682	0.015	0.030	0.166	7.34
7	0.065	0.052	0.679	0.015	0.038	0.151	9.87
8	0.067	0.028	0.676	0.015	0.034	0.180	7.31
9	0.068	0.052	0.685	0.016	0.037	0.142	2.68
10	0.068	0.032	0.709	0.013	0.032	0.146	19.44
11	0.069	0.037	0.724	0.013	0.031	0.128	12.91
12	0.070	0.055	0.662	0.014	0.038	0.161	3.72
13	0.070	0.044	0.667	0.017	0.038	0.164	5.19
14	0.071	0.045	0.680	0.014	0.029	0.161	13.02
15	0.072	0.046	0.667	0.015	0.045	0.156	3.82
16	0.072	0.052	0.704	0.017	0.039	0.117	9.60
17	0.072	0.052	0.660	0.017	0.032	0.168	10.34
18	0.072	0.038	0.674	0.014	0.036	0.165	4.22
19	0.073	0.028	0.641	0.017	0.036	0.205	8.56
20	0.073	0.051	0.666	0.014	0.036	0.162	10.80
21	0.075	0.037	0.688	0.015	0.033	0.152	14.16
22^*	0.075	0.033	0.654	0.014	0.027	0.197	4.64
23	0.075	0.053	0.630	0.016	0.036	0.190	7.32
24	0.075	0.039	0.702	0.014	0.036	0.134	11.68
25^*	0.076	0.045	0.736	0.014	0.030	0.099	19.76
26	0.078	0.035	0.655	0.017	0.033	0.183	6.27
27	0.079	0.040	0.675	0.014	0.033	0.160	4.28
28	0.082	0.047	0.696	0.014	0.039	0.123	3.92
29	0.083	0.042	0.668	0.016	0.029	0.161	3.98
30	0.087	0.048	0.661	0.016	0.036	0.152	6.28
31^*	0.088	0.051	0.698	0.016	0.035	0.112	5.08

Core Duo processor, 2.20 GHz, 3.5 GB RAM). It is unusual that the coordinate-exchange algorithm takes longer than the point-exchange algorithm. This is due to the fact that the coordinate-exchange algorithm has to check the 234 constraints defined by the convex hull. Verifying constraints in the point-exchange algorithm is not required because that algorithm only uses candidate points that satisfy all the constraints of the design problem.

The follow-up design we obtained in this fashion is shown in Table 6. It has a D-criterion value of 1.6557×10^{-132} which is substantially higher than the D-criterion value of the design in Table 3, 9.2665×10^{-138}. The design generated using the coordinate-algorithm is therefore 54.09% more efficient than that generated using the point-exchange algorithm with the candidate set involving the pure samples and the binary and ternary blends.

The D-optimal 31-run design we obtained using the coordinate-exchange algorithm is displayed in Table 7. That design has a D-criterion value of 9.1541×10^{-132}, so that it is 6.30% more efficient than the design in Table 5 and 63.72% more efficient than the design in Table 3.

6 Discussion

In the previous sections, we show that the coordinate-exchange algorithm used by JMP 8 outperforms the point-exchange algorithm used by SAS 9.2 when it comes to finding D-optimal designs for the baking experiment. This demonstrates the added value of the coordinate-exchange algorithm in highly constrained design regions. However, for the baking experiment, the designs produced by the coordinate-exchange algorithm are practically infeasible because they involve complicated blends of the 30 original samples. For these designs, it is a true challenge to determine which pure samples should be blended to obtain the design points. Also, the proportions for each of the blends are sometimes very unequal, which makes the blending nearly impossible for some of the design points. This is in contrast with the point-exchange algorithm, which yields designs that contain pure samples and binary and ternary blends with equal proportions only.

Another complication of the baking experiment is that the level of the process variable is uncontrollable. This means that it is a characteristic variable of the sample that cannot be altered. Blending samples thus means that also the value of this process variable is unavoidably set. For the point-exchange algorithm case this is intrinsically incorporated in the set of candidate points. In contrast, this practical constraint cannot be specified in the coordinate-exchange algorithm. Therefore, the designs obtained by the coordinate-exchange algorithm should be considered as a theoretical optimum, rather than a practical solution for the baking experiment. These designs show us what the statistical cost is of (i) the fact that only binary and ternary blends of the original samples can be used, and (ii) the fact that the level of the process variable cannot be set independently.

Table 6: Alternative design generated using the coordinate-exchange algorithm.

Type							
of Run	x_1	x_2	x_3	x_4	x_5	x_6	z
initial	0.058	0.052	0.654	0.014	0.037	0.185	10.14
initial	0.088	0.051	0.698	0.016	0.035	0.112	5.08
initial	0.061	0.028	0.712	0.012	0.031	0.156	6.06
initial	0.075	0.033	0.654	0.014	0.027	0.197	4.64
initial	0.064	0.027	0.644	0.015	0.035	0.216	3.44
initial	0.072	0.042	0.686	0.015	0.033	0.152	3.94
initial	0.076	0.045	0.736	0.014	0.030	0.099	19.76
follow-up	0.075	0.033	0.654	0.014	0.027	0.196	19.76
follow-up	0.065	0.051	0.678	0.015	0.038	0.153	19.76
follow-up	0.074	0.050	0.668	0.014	0.041	0.151	11.22
follow-up	0.075	0.052	0.632	0.016	0.036	0.189	19.76
follow-up	0.065	0.043	0.682	0.015	0.030	0.166	19.76
follow-up	0.063	0.044	0.680	0.013	0.035	0.165	2.68
follow-up	0.071	0.044	0.663	0.017	0.039	0.166	11.22
follow-up	0.076	0.040	0.701	0.014	0.036	0.133	2.68
follow-up	0.062	0.037	0.683	0.014	0.040	0.164	11.22
follow-up	0.072	0.045	0.666	0.015	0.044	0.157	2.68
follow-up	0.084	0.045	0.662	0.015	0.035	0.158	11.22
follow-up	0.072	0.051	0.660	0.017	0.032	0.167	2.68
follow-up	0.069	0.045	0.705	0.015	0.036	0.130	2.68
follow-up	0.072	0.029	0.642	0.017	0.036	0.204	19.76
follow-up	0.072	0.051	0.702	0.017	0.038	0.120	19.76
follow-up	0.072	0.045	0.666	0.015	0.044	0.157	19.76
follow-up	0.082	0.046	0.695	0.014	0.039	0.124	19.76
follow-up	0.075	0.032	0.648	0.017	0.034	0.194	2.68
follow-up	0.069	0.030	0.685	0.014	0.033	0.168	19.76
follow-up	0.082	0.041	0.668	0.016	0.029	0.162	19.76
follow-up	0.079	0.050	0.695	0.016	0.037	0.123	11.22
follow-up	0.076	0.052	0.633	0.016	0.036	0.187	2.68
follow-up	0.061	0.039	0.650	0.014	0.036	0.200	19.76
follow-up	0.072	0.052	0.667	0.014	0.036	0.159	19.76

Table 7: D-optimal 31-run design generated using the coordinate-exchange algorithm.

Run	x_1	x_2	x_3	x_4	x_5	x_6	z
1	0.065	0.051	0.679	0.015	0.038	0.151	19.76
2	0.079	0.044	0.699	0.014	0.038	0.126	11.22
3	0.087	0.048	0.663	0.016	0.036	0.151	2.68
4	0.072	0.051	0.702	0.017	0.038	0.120	11.22
5	0.080	0.041	0.651	0.016	0.036	0.176	19.76
6	0.076	0.045	0.733	0.014	0.030	0.103	2.68
7	0.058	0.051	0.655	0.014	0.037	0.184	2.68
8	0.075	0.047	0.670	0.015	0.043	0.150	19.76
9	0.061	0.029	0.709	0.012	0.031	0.156	19.76
10	0.064	0.027	0.646	0.015	0.034	0.214	11.22
11	0.087	0.051	0.697	0.015	0.035	0.115	19.76
12	0.076	0.045	0.733	0.014	0.030	0.103	19.76
13	0.081	0.051	0.697	0.016	0.037	0.119	2.68
14	0.070	0.030	0.679	0.014	0.033	0.173	19.76
15	0.063	0.038	0.651	0.015	0.035	0.198	19.76
16	0.075	0.033	0.655	0.014	0.027	0.195	2.68
17	0.075	0.052	0.632	0.016	0.037	0.188	11.22
18	0.072	0.045	0.666	0.015	0.044	0.157	2.68
19	0.073	0.029	0.642	0.017	0.036	0.204	19.76
20	0.075	0.033	0.653	0.014	0.027	0.197	19.76
21	0.072	0.051	0.660	0.017	0.032	0.167	2.68
22	0.072	0.051	0.659	0.017	0.032	0.169	19.76
23	0.073	0.053	0.669	0.014	0.038	0.154	2.68
24	0.062	0.029	0.707	0.012	0.031	0.158	2.68
25	0.065	0.043	0.680	0.015	0.030	0.167	2.68
26	0.070	0.051	0.673	0.017	0.041	0.148	2.68
27	0.073	0.029	0.642	0.017	0.036	0.204	2.68
28	0.071	0.045	0.683	0.014	0.029	0.157	11.22
29	0.062	0.037	0.682	0.014	0.040	0.165	2.68
30	0.058	0.051	0.655	0.014	0.037	0.184	19.76
31	0.082	0.041	0.668	0.016	0.029	0.162	11.22

The design that was actually used to conduct the baking experiment is displayed in Table 3. With the data from the experiment (which are proprietary), the model in Equation (4) was estimated. The analysis yielded surprising results, and suggested a way to improve the bread quality substantially. A new follow-up experiment was then conducted to confirm the earlier finding and to verify whether the quality improvement was reproducible. For this reason, the follow-up experiment focused on a smaller design region. The design of the follow-up experiment posed a new design challenge: the available amount of flour for some of the original 30 samples was very low. A new type of constraint, the limited availability of certain samples, therefore had to be taken into account when designing this new follow-up study. To the best of our knowledge, this type of constraint has not received attention yet in the literature and forms an interesting subject for future research.

A practical issue in the baking experiment was that the week-to-week variation was monitored using a weekly reference sample. While this is a very sensible thing to do and, hence, common practice, this is not the most informative strategy. We would generally recommend not to use reference samples. Instead, we would design an experiment with one extra run per week, include block effects in the model and treat these block effects as random effects. The resulting estimates for the model parameters would be more precise than in the approach involving the reference samples.

7 Summary

In this article, we describe a real-life baking experiment and suggest a procedure to select an ideal set of factor level combinations starting from an available stock of flour samples for estimating a mixture-process variable model. By ideal, we mean here that the selected samples should allow for the most precise estimation of the parameters of a MPV model. We briefly reviewed the traditional Scheffé type of models for mixture variables and their combination with response surface models for the process variables. In general, MPV experiments can be designed by combining mixture designs with process-variable designs. When the design region is constrained as in the baking experiment, a computerized optimal design approach is needed. We explain how this design approach was used to create a design for the baking experiment, and compare two different strategies, one based on a point-exchange algorithm and another based on a coordinate-exchange algorithm. The coordinate-exchange algorithm, which requires the determination of the convex hull of the initial stock of flour samples, yields the most efficient designs, but they are practically infeasible. In contrast, the point-exchange algorithm starts from a limited candidate set, and results in a design that is easy to implement in the baking experiment. The drawback of using the point-exchange algorithm is that it is substantially less efficient than that produced by the coordinate-exchange algorithm. The practicalities of the experiment, however, dictated that a design produced by the point-exchange algorithm had to be used.

Appendix A. SAS code for implementing the point exchange algorithm

The input of the initial design requires the following input:

```
data initialset;
input x1 x2 x3 x4 x5 x6 z;
datalines;
0.061 0.028 0.712 0.012 0.031 0.156  6.06
...
0.058 0.052 0.654 0.014 0.037 0.185 10.14
;
run;
```

The input of the candidate set requires the following input:

```
data candidateset;
input x1 x2 x3 x4 x5 x6 z;
datalines;
0.074 0.042 0.679 0.015 0.032 0.158 11.92
...
;
run;
```

Finding the D-optimal follow-up design for the baking experiment then requires calling the OPTEX procedure using the following commands:

```
proc optex data=candidateset;
model  x1|x2|x3|x4|x5|x6|z@2 z*z / noint;
generate n = 31 augment = initialset iter = 5000;
output out = Doptim;
run;
```

The model specified in the "model" statement is the 28-parameter model given in Equation (4). The absence of an intercept in that model requires specifying the "noint" option in the "model" statement. The "generate" statement is used to define the total number of observations in the design. For the baking experiment, this number is 31 (seven observations in the initial design and 24 in the follow-up experiment). In the "generate" statement, we also specify the data set containing the initial design. The "output" statement ensures that the D-optimal design is saved in a data set with the name "Doptim". The exact point-exchange algorithm that is used by these commands is the k-exchange algorithm described in Johnson and Nachtsheim (1983), which is the default in SAS.

Appendix B. Script for calculating the constraints defined by the convex hull

We performed the calculation of the convex hull encompassing the 6 mixture variables x_1-x_6 of the 30 samples shown in Table 1 using the MATLAB function “convhulln”. It uses the quickhull algorithm described by Barber et al. (1996) and returns the indices k of the points in the input data matrix that comprise the facets of the convex hull:

```
k = convhulln(X);
```

The $\mathbf{X}$ matrix has as dimensions (30 by 5) reflecting each of the 30 samples as rows. The fact that there are only 5 columns whilst there are 6 mixture variables is related to the constraint that all mixture components sum to 1, lowering the dimensionality of the problem by 1. Hence, we included only x_1-x_5 to form $\mathbf{X}$. Remark that the process variable z is not related to the convex hull computation – it forms the subject of another set of constraints, as we explained above. The output of the “convhulln” command denoted by k is a 234 by 5 matrix in our baking experiment. The number 234 reflects the number of facets of the convex hull. Each facet (row in k) is defined by the 5 samples that lay on the facet.

The 234 constraints that will be used in the coordinate-exchange algorithm are derived by setting up the (hyper-)plane equation of each of the facets, and also the corresponding side of the plane that points towards the centre. The equation of a plane in the 5 dimensions has the form

$$a_0 + a_1 v + a_2 w + a_3 x + a_4 y + a_5 z = 0,$$

The plane coefficients a_0 to a_5 can be found using the coordinates of at least 5 points that lay on the plane. Practically, the points defining each facet of the convex hull (i.e. each row in k) are used, and the set of 5 equations given as:

$$a_0 + a_1 v_1 + a_2 w_1 + a_3 x_1 + a_4 y_1 + a_5 z_1 = 0,$$

$$a_0 + a_1 v_2 + a_2 w_2 + a_3 x_2 + a_4 y_2 + a_5 z_2 = 0,$$

$$a_0 + a_1 v_3 + a_2 w_3 + a_3 x_3 + a_4 y_3 + a_5 z_3 = 0,$$

$$a_0 + a_1 v_4 + a_2 w_4 + a_3 x_4 + a_4 y_4 + a_5 z_4 = 0,$$

$$a_0 + a_1 v_5 + a_2 w_5 + a_3 x_5 + a_4 y_5 + a_5 z_5 = 0,$$

is solved using Cramer’s rule. Since there are 6 parameters and only 5 equations, the solution is not unique. Setting a_0 equal to any non-zero number and substituting it into these equations will yield one solution set.

The following MATLAB script can be used to perform the above calculations for the dataset $\mathbf{X}$ considered in this paper:

```
m = mean(X);
dim = 5;
a0 = ones(length(k(:,1)),1);
a1 = zeros(length(k(:,1)),1);
a2 = zeros(length(k(:,1)),1);
a3 = zeros(length(k(:,1)),1);
a4 = zeros(length(k(:,1)),1);
a5 = zeros(length(k(:,1)),1);
for i = 1:length(k(:,1)),
    v = X(k(i,1:dim),1);
    w = X(k(i,1:dim),2);
    x = X(k(i,1:dim),3);
    y = X(k(i,1:dim),4);
    z = X(k(i,1:dim),5);
    D(i) = det([v w x y z]);
    a1(i) = -(a0(i)/D(i))*det([ones(dim,1) w x y z]);
    a2(i) = -(a0(i)/D(i))*det([v ones(dim,1) x y z]);
    a3(i) = -(a0(i)/D(i))*det([v w ones(dim,1) y z]);
    a4(i) = -(a0(i)/D(i))*det([v w x ones(dim,1) z]);
    a5(i) = -(a0(i)/D(i))*det([v w x y ones(dim,1)]);
    s(i) = sign(a0(i)+a1(i)*m(1)+a2(i)*m(2)+a3(i)*m(3)+a4(i)*m(4)+a5(i)*m(5));
end;
```

where a_0 to a_5 are the plane coefficients (with a_0 set to 1 in the example) and $\boldsymbol{m}$ denoting the mean vector of the mixture variables x_1 to x_5. This mean vector is used to define the side of the plane that points towards the centre of the convex hull. The variable s in the MATLAB code determines, for each constraint i, whether it is a $\leq$ constraint or a $\geq$ constraint. A negative sign s is equivalent to

$$a_0 + a_1 v + a_2 w + a_3 x + a_4 y + a_5 z \leq 0,$$

or, in our practical case with $a_0 = 1$, to

$$a_1 v + a_2 w + a_3 x + a_4 y + a_5 z \leq -1.00.$$

A positive sign s then dictates

$$a_0 + a_1 v + a_2 w + a_3 x + a_4 y + a_5 z \geq -1.00,$$

or, equivalently,

$$-a_0 - a_1 v - a_2 w - a_3 x - a_4 y - a_5 z \leq 1.00.$$

Appendix C. Script for problem with 234 constraints

The generation of the D-optimal 31-run mixture-process variable design using JMP 8 and the 234 constraints defined by the convex hull of the 30 original samples can be done using a script. The advantage of this approach is that the constraints need not be inputted manually one by one. In the first part of the script, six mixture variables (x_1-x_6) are created, along with one continuous process variable (z) which can take values between 2.68 and 19.76 only. Next, the number of random starts of the algorithm is fixed and the constraints are listed.

Each constraint corresponds to one row of eight values. Seven of the values correspond to the experimental variables. The eighth value corresponds to the right-hand side of the constraint. We generated the values in each row so that the constraint is a $\leq$ constraint. For example, the first constraint in our design problem is

$$-2.50x_1 + 0.59x_2 - 1.15x_3 + 3.93x_4 - 4.97x_5 + 0.00x_6 + 0.00z \leq -1.00,$$

which can be simplified to

$$-2.50x_1 + 0.59x_2 - 1.15x_3 + 3.93x_4 - 4.97x_5 \leq -1.00,$$

Regarding the constraints, it is worth noting that the coefficients of x_6 and z are always zero. For z, this is simply due to the fact that the process variable is not involved in any of the constraints. For x_6, the reason is different. In principle, every constraint does involve x_6, but the mixture constraint

$$\sum_{i=1}^{6} x_i = 1$$

ensures that each of the constraints can be rewritten so that any of the mixture variables x_i has a zero coefficient.

In the last part of the JMP script, the model is specified, as well as the desired number of runs.

The script we used is as follows:

```
DOE(
Custom Design,
{Add Response( Maximize, "Y", ., ., . ),
 Add Factor( Mixture, 0, 1, "X1", 0 ),
 Add Factor( Mixture, 0, 1, "X2", 0 ),
 Add Factor( Mixture, 0, 1, "X3", 0 ),
 Add Factor( Mixture, 0, 1, "X4", 0 ),
 Add Factor( Mixture, 0, 1, "X5", 0 ),
 Add Factor( Mixture, 0, 1, "X6", 0 ),
 Add Factor( Continuous, 2.68, 19.76, "Z", 0 ),
 Set Random Seed( 17113366 ),
 Number of Starts( 10 ),
 Add Constraint(
[
 -2.50    0.59   -1.15     3.93    -4.97   0.00   0.00   -1.00,
  1.46   -0.77   -1.02   -34.49     2.51   0.00   0.00   -1.00,
  ...
  4.79   -8.54    1.14   -36.81    18.73   0.00   0.00    1.00,
  4.74   -8.47    1.13   -35.15    18.41   0.00   0.00    1.00,
  5.34   -9.42    1.10   -37.21    19.50   0.00   0.00    1.00
      ]
),
 Add Term( {1, 1} ), Add Term( {2, 1} ), Add Term( {3, 1} ),
 Add Term( {4, 1} ), Add Term( {5, 1} ), Add Term( {6, 1} ),
 Add Term( {1, 1}, {2, 1} ), Add Term( {1, 1}, {3, 1} ),
 Add Term( {1, 1}, {4, 1} ), Add Term( {1, 1}, {5, 1} ),
 Add Term( {1, 1}, {6, 1} ), Add Term( {1, 1}, {7, 1} ),
 Add Term( {2, 1}, {3, 1} ), Add Term( {2, 1}, {4, 1} ),
 Add Term( {2, 1}, {5, 1} ), Add Term( {2, 1}, {6, 1} ),
 Add Term( {2, 1}, {7, 1} ), Add Term( {3, 1}, {4, 1} ),
 Add Term( {3, 1}, {5, 1} ), Add Term( {3, 1}, {6, 1} ),
 Add Term( {3, 1}, {7, 1} ), Add Term( {4, 1}, {5, 1} ),
 Add Term( {4, 1}, {6, 1} ), Add Term( {4, 1}, {7, 1} ),
 Add Term( {5, 1}, {6, 1} ), Add Term( {5, 1}, {7, 1} ),
 Add Term( {6, 1}, {7, 1} ), Add Term( {7, 2} ),
 Set Sample Size( 31 )}
);
```

References

Anderson-Cook, C. M. and Robinson, T. J. (2009). A designed screening study with prespecified combinations of factor settings, *Quality Engineering* **21**: 392–404.

Atkinson, A. C. and Donev, A. N. (1989). The construction of exact D-optimum experimental designs with application to blocking response surface designs, *Biometrika* **76**: 515–526.

Barber, C. B., Dobkin, D. P. and Huhdanpaa, H. T. (1996). The Quickhull algorithm for convex hulls, *ACM Transactions on Mathematical Software* **22**: 469–483.

Box, G. E. P., Hunter, W. and Hunter, J. (1978). *Statistics for Experimenters*, New York: Wiley.

Cook, R. D. and Nachtsheim, C. J. (1989). Computer-aided blocking of factorial and response-surface designs, *Technometrics* **31**: 339–346.

Cornell, J. A. (1988). Analyzing data from mixture experiments containing process variables: A split-plot approach, *Journal of Quality Technology* **20**: 2–23.

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

Cornell, J. A. and Gorman, J. W. (1984). Fractional design plans for process variables in mixture experiments, *Journal of Quality Technology* **16**: 20–38.

Czitrom, V. (1988). Mixture experiments with process variables: D-optimal orthogonal experimental plans, *Communications in Statistics: Theory and Methods* **17**: 105–121.

Czitrom, V. (1989). Experimental designs for four mixture components with process variables, *Communications in Statistics: Theory and Methods* **18**: 4561–4581.

Dingstad, G., Egelandsdal, B. and Naes, T. (2003). Modeling methods for crossed mixture experiments: A case study from sausage production, *Chemometrics and Intelligent Laboratory Systems* **66**: 175–190.

Fedorov, V. V. (1972). *Theory of Optimal Experiments*, New York: Academic Press.

Goos, P. and Donev, A. N. (2007). Tailor-made split-plot designs with mixture and process variables, *Journal of Quality Technology* **39**: 326–339.

Johnson, M. E. and Nachtsheim, C. J. (1983). Some guidelines for constructing exact D-optimal designs on convex design spaces, *Technometrics* **25**: 271–277.

Kowalski, S. M., Cornell, J. A. and Vining, G. G. (2000). A new model and class of designs for mixture experiments with process variables, *Communications in Statistics: Theory and Methods* **29**: 2255–2280.

Kowalski, S. M., Cornell, J. A. and Vining, G. G. (2002). Split-plot designs and estimation methods for mixture experiments with process variables, *Technometrics* **44**: 72–79.

Mage, I. and Naes, T. (2005). Split-plot design for mixture experiments with process variables: A comparison of design strategies, *Chemometrics and Intelligent Laboratory Systems* **78**: 81–95.

Meyer, R. K. and Nachtsheim, C. J. (1995). The coordinate-exchange algorithm for constructing exact optimal experimental designs, *Technometrics* **37**: 60–69.

Nachtsheim, C. J. (1989). On the design of experiments in the presence of fixed covariates, *Journal of Statistical Planning and Inference* **22**: 203–212.

Naes, T., Aastveit, A. H. and Sahni, N. S. (2007). Analysis of split-plot designs: an overview and comparison of methods, *Quality and Reliability Engineering International* **23**: 801–820.

Naes, T., Faergestad, E. M. and Cornell, J. (1998). A comparison of methods for analyzing data from a three component mixture experiment in the presence of variation created by two process variables, *Chemometrics and Intelligent Laboratory Systems* **41**: 221–235.

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.

Prescott, P. (2004). Modelling in mixture experiments including interactions with process variables, *Quality Technology and Quality Management* **1**: 87–103.

Rehman, S., Paterson, A. and Piggott, J. R. (2007). Optimisation of flours for chapatti preparation using a mixture design, *Journal of the Science of Food and Agriculture,* **87**: 425–430.

Sexton, C. J., Anthony, D. K., Lewis, S. M., Please, C. P. and Keane, A. J. (2006). Design of experiments algorithms for assembled products, *Journal of Quality Technology* **38**: 298–308.

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