Optimal experimental design in the presence of nested factors

Optimal Experimental Design in the Presence of Nested Factors

Peter Goos & Bradley Jones

View supplementary material

Accepted author version posted online: 24 Jan 2019.

Submit your article to this journal

Article views: 11

View Crossmark data

Optimal Experimental Design in the Presence of Nested Factors

Peter Goos

Faculty of Bioscience Engineering and Leuven Statistics Research Centre

KU Leuven

Faculty of Applied Economics and StatUa Center for Statistics

Universiteit Antwerpen

Bradley Jones

SAS Institute

Abstract–**A common occurrence in practical design of experiments is that one factor, called a nested factor, can only be varied for some but not all the levels of a categorical factor, called a branching factor. In this case, it is possible, but inefficient, to proceed by performing two experiments. One experiment would be run at the level(s) of the branching factor that allow for varying the second, nested, factor. The other experiment would only include the other level(s) of the branching factor. It is preferable to perform one experiment that allows for assessing the effects of both factors. Clearly, the effect of the nested factor then is conditional on the levels of the branching factor for which it can be varied. For example, consider an experiment comparing the performance of two machines where one machine has a switch that is missing for the other machine. The investigator wants to compare the two machines but also wants to understand the effect of flipping the switch. The main effect of the switch is conditional on the machine. This article describes several example situations involving branching factors and nested factors. We provide a model that is sensible for each situation, present a general method for constructing appropriate models, and show how to generate optimal designs given these models. Supplementary materials for this article are available online.**

Keywords: branching factor, conditional main effect, coordinate-exchange algorithm, D-optimal design, nested factor, point-exchange algorithm, shared factor, sliding levels.

1 Introduction

For many experiments, certain factors are only relevant for given levels of another factor. Hung et al. (2009) call these factors nested factors. They call factors within which other factors are nested branching factors, and all other

factors are shared factors. Branching factors are categorical, while nested and shared factors are either quantitative or categorical. For example, consider an experiment intended to compare the performance of two machines at different speeds, where one machine has a switch that is missing for the other machine. Besides quantifying the effect of speed, the investigator wants to compare the two machines as well as understand the effect of flipping the switch. In that case, the machine is a two-level branching factor, while the switch is a two-level nested factor. The nested factor only exists at one of the branching factor's levels. The third factor, speed, is a shared factor. In this example, the main effect of the switch is conditional on the machine.

Recently, several papers have been published on the subject of conditional main effects. A conditional main effect is the main effect of one factor while fixing the level of another factor. Wu (2015) introduces conditional main effects as a way of circumventing the confounding of two-factor interactions in resolution IV fractional factorial designs. Su and Wu (2017) provide a set rules for finding an adequate set of conditional main effects in the analysis of fractional factorial designs. They also provide three illustrative examples. Mak and Wu (2018) extend the work on model selection for conditional main effects to observational studies. The concept of a conditional main effect is similar to that of a simple (main) effect in the context of multi-way analysis of variance, which is defined as the effect of one factor within one level of a second factor. In analysis of variance, simple effects are generally estimated after discovering a significant interaction effect, and facilitate the understanding of the nature of the interaction. A key feature of the work of Wu (2015), Su and Wu (2017) and Mak and Wu (2018) is that their use of conditional main effects is a means to build better models for given data sets. It is not inspired by the existence of branching factors and nested factors.

In this article, we also consider conditional main effects. However, a major distinction between our work and that of Wu (2015), Su and Wu (2017) and Mak and Wu (2018) is that we focus on the design of experiments where it is known *a priori* which conditional main effects should be included in the model, due to the fact that it is known *a priori* which factors are nested within a

branching factor. We also go beyond conditional main effects and deal with conditional quadratic effects and conditional interaction effects. The relevance of our work is supported by experiments discussed in the literature. Hung et al. (2009), for example, discuss a computer experiment involving a two-level branching factor, with two nested factors. The nested factors only exist for one level of the branching factor. In addition to the branching factor and the two nested factors, the experiment also involved six shared factors. Schoen et al. (2011) discuss an experiment involving a two-level branching factor and a nested factor, whose levels depend on the level of the branching factor. Finally, Palmers et al. (2016) discuss an experiment involving a branching factor, two nested factors and conditional main effects, quadratic effects and interaction effects.

We illustrate the need for a generic method to cope with conditional effects of any kind with several real-world examples. The first example concerns a chicken feed experiment. The second example involves an extraction experiment in the pharmaceutical industry. The third example is the magnetic bead experiment in Decrop et al. (2016, 2017).

The rest of this article is organized as follows. In Section 2, we consider a small proof-of-concept example, provide D-optimal designs for different numbers of runs and compare their efficiencies with those of replicated 2 × 2 factorial designs. In Section 3, we present a generic method to build suitable statistical models for any number of branching factors, nested factors and shared factors, involving conditional main effects, conditional quadratic effects and conditional interaction effects. In Section 4, we discuss the chicken feed experiment, the extraction experiment and the magnetic bead experiment in detail. In Section 5, we explain how to generate D-optimal designs. More specifically, we explain that point-exchange algorithms can be used and we present a modified coordinate-exchange algorithm to deal with branching factors and nested factors. In Section 6, we compare D-optimal designs to standard designs for two of the examples in Su and Wu (2017), assuming that the relevance of certain conditional main effects is known in advance due to the nature of the physical system under investigation. We do this to show that,

if a conditional main effect can be identified in advance, then it pays to use this information when generating the experimental design. We conclude with a discussion in Section 7.

2 A proof-of-concept example

Suppose we wish to compare the performance of a newer and an older machine, where the newer has a dial controlling the level of a quantitative factor. To this end, we can perform an experiment with two factors, the machine and the quantitative factor. The machine is the branching factor and the quantitative factor is the nested factor. Given that the quantitative factor can only be studied for the new machine, its effect is conditional on the new machine being used. Assuming that the dial factor's effect is linear, a useful model is

$$Y = \beta_0 + \beta_1 z + \beta_2 z x + \varepsilon, \qquad (1)$$

where z is an indicator variable taking the value 1 if the new machine is used and the value 0 otherwise, and x is the setting of the quantitative factor. The symbols Y, ε and β_i represent the response, the error term and the regression coefficients in this model and all subsequent ones. A key feature of the model in Equation (1) is that the nested factor, x, only enters the model through its interaction term with the indicator variable z, which provides information on the level of the branching factor.

Figure 1 visualizes the design problem in the proof-of-concept example. The indicator variable z takes the value 1 when the new machine is used. In that case, the dial factor needs to be set to a value from the $[-1, +1]$ interval.

A design strategy that might be considered is to spend half of the experimental resources on the older machine and other half of the resources on the newer machine. For the older machine, the second factor is irrelevant, since it does not have a dial. For the second machine, the dial factor could be set to a low level half the time and to a high level the other half of the time. This results in the design in Table 1.

The design strategy shown in Table 1 is equivalent to starting from a multiply replicated 2^2 factorial design, and ignoring the level of the second factor either when the first factor is at its low level or when the first factor is at its high level. While this strategy is simple and results in independent estimates of β_0, β_1 and β_2, it leads to a standard error for β_2 which is substantially higher than that for β_1. This is shown in the left panel of Table 2, labeled $r \times 2^2$, for two replicates (r = 2), three replicates (r = 3) and four replicates (r = 4). Note that the standard errors in the table are relative to the standard deviation of ε. Generally, we prefer all factor effects to be estimated as precisely as possible, in an omnibus sense. One possibility to lower the standard error for β_2 is to create optimal experimental designs for the model in Equation (1).

Optimal experimental designs ensure either precise model estimation or precise predictions. The first commercially available kinds of optimal designs were D-optimal designs, which focus on precise model estimation and minimize the generalized variance of the estimates of the regression coefficients. There are several alternative definitions of the D-optimality criterion. In this paper, we use the determinant of the information matrix, which summarizes all information on the regression coefficients contained within the experiment, as the D-optimality criterion. A design that maximizes that criterion is D-optimal. Another criterion that focuses on precise model estimation is the A-optimality criterion, which aims to minimize the average variance of the parameter estimates. In this article, we report the performances of the various designs using D- and A-efficiencies. D- and A-efficiencies lie between 0 and 1, and quantify how well designs perform relative to a theoretically optimal design that is orthogonal and yields maximum information on each model parameter. Note that such a theoretically optimal design never exists for designs with nested factors. As a result, reported efficiencies represent a loose lower bound on the true efficiency. Detailed technical information concerning the D- and A-optimality criteria can be found in Atkinson et al. (2007) and Goos and Jones (2011). In this article, we compute D-optimal designs and compare them to standard benchmark designs in terms of D- and A-efficiency.

The top part of Table 2 contrasts the D-optimal designs involving 8, 12 and 16 runs for the model in Equation (1) with the designs based on the replicated 2^2 factorial design. More specifically, the table shows the three feasible test combinations along with the number of times they appear in the D-optimal experimental designs and in the benchmark designs.

The most striking feature of the D-optimal designs is that the two machines are not used equally often. Whenever the number of runs is a multiple of 3, there is only one D-optimal design: the old machine then appears in one third of the runs, while the new machine appears in the other two thirds of the runs. Half of the runs involving the new machine require the dial to be set at its low level, while the other half require the dial to be set at its high level. This is demonstrated by the 12-run D-optimal design in Table 2. When the number of runs is not a multiple of 3, the D-optimal design will be as close to being balanced as possible, and there will generally be more than one D-optimal design.

Since 8 and 16 are not divisible by 3, it is not possible to use each test combination equally often. Which test combination is used the most/least often does not impact the D-optimality of the design. However, an intuitively appealing option is to ensure that the low and high levels of the dial factor are equireplicated. Table 2 shows two 8-run D-optimal designs as well as two 16-run D-optimal designs, one in which the low and high levels of the dial factor are equireplicated and one in which they are not. We labeled the two alternatives I and II in Table 2.

The standard errors resulting from the D-optimal designs are displayed in the bottom-right panel of Table 2. Comparing the standard errors in the left and right panels of the table shows that the standard error for β_2 improves substantially when a D-optimal design is used, while the standard errors for β_0 and β_1 increase. The last two lines of Table 2 compare the D- and A-efficiencies of all design options, and show that the D-efficiencies of the D-optimal designs are superior to those of the alternative designs and that the

A-efficiencies of the D-optimal designs are larger than or equal to those of the alternative designs.

That the old and the new machine are not used equally often by the D-optimal design may seem counterintuitive. To understand why it is desirable to use the new machine more often, one must realize that the only runs providing information about the dial factor's conditional main effect are those employing the new machine. The D-optimal design trades a small loss in the precision of the estimate of the machine effect for an appreciable gain in the precision of the estimate of the dial factor's effect.

In the event a quadratic effect of the dial factor is anticipated, the *a priori* model becomes

$$Y = \beta_0 + \beta_1 d + \beta_2 zx + \beta_3 zx^2 + \varepsilon, \qquad (2)$$

and the D-optimal designs change as well. For the model involving the quadratic dial factor effect, ideally, a quarter of the runs use the old machine, while three quarters of the runs involve the new machine. One third of the runs with the new machine require setting the dial factor to its low level. Another third of the runs with the new machine require setting the dial factor to its high level, and the final third of the runs with the new machine require setting the dial factor to its middle level. Whenever the number of runs is not a multiple of four, the four test combinations of the D-optimal design need to be as equireplicated as possible.

3 Model building

In the proof-of-concept example in the previous section, there was one two-level categorical branching factor. However, in some experiments, the branching factor is a categorical factor with more than two levels. Sometimes, investigators are concerned about potential curvature in the effects of the nested factors, which necessitates the estimation of multiple conditional quadratic effects. We also encountered experiments in which a nested factor may have two-way interactions with other nested factors or with shared

factors, which gives rise to conditional interaction effects. Finally, we also came across scenarios in which, rather than the existence, it is the range of a nested factor which depends on the level of a branching factor.

It is often challenging to write down a suitable, identifiable model for experiments with branching factors, nested factors and shared factors. Yet, specifying an identifiable model is a key requirement for constructing an optimal experimental design. This is because the information matrix corresponding to a non-identifiable model is always singular, regardless of the experimental design chosen.

The first step required to build a suitable model is to identify the shared factors in the experiment and to write down the *a priori* model for these factors. This can be a main-effects model, a main-effects-plus-interactions model or a second-order response surface model. In this section, we assume there are s quantitative shared factors and that the interest is in their main effects, two-factor interaction effects and quadratic effects. The initial model then is

$$Y = \beta_0 + \sum_{i=1}^{s} \beta_{(s)i} x_{(s)i} + \sum_{i=1}^{s-1} \sum_{j=i+1}^{s} \beta_{(s)ij} x_{(s)i} x_{(s)j} + \sum_{i=1}^{s} \beta_{(s)ii} x_{(s)i}^2 + \varepsilon, \quad (3)$$

where $x_{(s)i}$ denotes the level of the ith shared factor and $\beta_{(s)i}, \beta_{(s)ij}$ and $\beta_{(s)ii}$ represent the main effects, two-factor interaction effects and quadratic effects, respectively, of the shared factors. In some experiments, such as in our proof-of-concept example in Section 2 and the chicken feed experiment and the extraction experiment in Section 4, there are no shared factors, so that $s = 0$ and the initial model only involves an intercept parameter β_0.

The second step is to identify the branching factors and the nesting relationships associated with them. At most one nesting relationship corresponds with each level of a branching factor. Each nesting relationship may involve one or more nested factors and gives rise to one indicator variable. An indicator variable corresponding to the ith level of a branching factor has the following functions:

- it should modify the model's intercept in the event the kth level of the branching factor is selected;
- it should create model terms for the conditional main effects of all nested factors associated with the kth level of the branching factor;
- it may also create model terms for the conditional quadratic effects of all nested factors associated with the kth level of the branching factor;
- it may also create model terms for the conditional interaction effects involving one or more nested factors associated with the kth level of the branching factor;
- it should ensure that the conditional effects of the nested factors associated with the kth level of the branching factor are eliminated from the model in the event the kth level of the branching factor is not selected.

The next step is to add the conditional effects of the nested factors to the initial model. *A key insight here is that any nested factor should only enter the model by means of a cross-product with the associated indicator variable.* The simplest terms to add are the conditional main effects and the conditional quadratic effects of the nested factors. If we denote the number of nesting relationships by κ and the λ nested factors associated with the kth nesting relationship by $x_{(n)k1}, x_{(n)k2}, \ldots, x_{(n)k\lambda}$, then the model becomes

$$
\begin{aligned}
Y \quad &= \beta_0 + \sum_{i=1}^{s} \beta_{(s)i} x_{(s)i} + \sum_{i=1}^{s-1} \sum_{j=i+1}^{s} \beta_{(s)ij} x_{(s)i} x_{(s)j} + \sum_{i=1}^{s} \beta_{(s)ii} x_{(s)i}^2 \\
&\quad + \sum_{k=1}^{\kappa} z_k \left(\sum_{i=1}^{\lambda} \beta_{(n)ki} x_{(n)ki} + \sum_{i=1}^{\lambda} \beta_{(n)kii} x_{(n)ki}^2 \right) + \varepsilon,
\end{aligned}
\tag{4}
$$

where z_k is the indicator variable corresponding to the kth nesting relationship. If that variable takes the value 1, then the corresponding nested factors, $x_{(n)k1}, x_{(n)k2}, \ldots, x_{(n)k\lambda}$, appear in the model. If it takes the value 0, then the corresponding nested factors do not appear in the model.

In the same fashion, interactions involving any two nested factors can be added. This requires the following extra term to enter the model:

$$\sum_{k=1}^{\kappa} z_k \sum_{i=1}^{\lambda-1} \sum_{j=i+1}^{\lambda} \beta_{(n)kij} x_{(n)ki} x_{(n)kj}.$$

In the event we want to allow for interactions involving the shared factors, on the one hand, and the nested factors, on the other hand, we also need to add the following term:

$$\sum_{k=1}^{\kappa} z_k \sum_{i=1}^{s} \sum_{j=1}^{\lambda} \beta_{(sn)kij} x_{(s)i} x_{(n)kj}.$$

Finally, to allow for branching level-specific intercepts, linear terms in the indicator variables,

$$\sum_{k=1}^{\kappa} \beta_k z_k,$$

have to be added. There is one case in which we cannot add all linear terms in the indicator variables: if an l-level branching factor has l nesting relationships, then, to avoid collinearity, we should drop one of the l associated indicator variable terms from the model. An example of such a case is the extraction experiment discussed in the next section.

In conclusion, a general model to deal with s shared factors, one or more branching factors involving κ nesting relationships, and λ nested factors corresponding to each nesting relationship, is given by

$$
\begin{aligned}
Y \quad &= \beta_0 + \sum_{k=1}^{\kappa} \beta_k z_k + \sum_{i=1}^{s} \beta_{(s)i} x_{(s)i} + \sum_{i=1}^{s-1} \sum_{j=i+1}^{s} \beta_{(s)ij} x_{(s)i} x_{(s)j} + \sum_{i=1}^{s} \beta_{(s)ii} x_{(s)i}^2 \\
&+ \sum_{k=1}^{\kappa} z_k \left(\sum_{i=1}^{\lambda} \beta_{(n)ki} x_{(n)ki} + \sum_{i=1}^{\lambda} \beta_{(n)kii} x_{(n)ki}^2 \right) + \sum_{k=1}^{\kappa} z_k \sum_{i=1}^{\lambda-1} \sum_{j=i+1}^{\lambda} \beta_{(n)kij} x_{(n)ki} x_{(n)kj} \quad (5) \\
&+ \sum_{k=1}^{\kappa} z_k \sum_{i=1}^{s} \sum_{j=1}^{\lambda} \beta_{(sn)kij} x_{(s)i} x_{(n)kj} + \varepsilon.
\end{aligned}
$$

A special feature of this model is that it encompasses a second-order response surface model in the s shared factors and the λ nested factors for each of the κ nesting relationships, as well as a second-order response

surface model in the s shared factors only for the scenario in which all indicator variables take the value zero.

Obviously, this model can be adapted to include categorical shared factors, categorical nested factors, and/or higher-order effects of any quantitative shared or nested factors. Also, the model can be adapted to cope with a problem in which the number of nested factors differs across the κ nesting relationships. In any case, it should be clear that the model complexity increases rapidly with the number of shared factors and the number(s) of nested factors. This is due to the fact that our model allows the nested factors to have effects that are specific to the level of the branching factor with which they are associated.

4 Practical examples

In this section, we describe several problems we encountered in our consulting. All problems are substantially more complex than the proof-of-concept example in Section 2. We discuss the kinds of regression models required to deal with the various complications, and break the models down to interpret them. Based on the models, it is then possible to compute optimal experimental designs.

4.1 A chicken feed experiment

4.1.1 Problem

In a chicken feed experiment carried out at the Faculty of Bioscience Engineering of the KU Leuven in Belgium, the effect of adding one or the other of two proposed enzymes to the feed was studied. Each of the enzymes was used multiple times at each of three different doses (10 ppm, 100 ppm and 1000 ppm). The experiment also included multiple tests involving a control treatment, which did not use either enzyme. One research question was whether the enzymes improved the growth. A second research question was whether there was a dose effect, and a third question was whether the dose effect differed between the two enzymes. Clearly, in the chicken feed

experiment, the branching factor, enzyme, is categorical, and the nested factor, dose, is numeric.

4.1.2 Model

Answering the three research questions in the chicken feed experiment requires fitting a suitable model and performing appropriate significance tests. A sensible model might be

$$Y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \beta_{11} z_1 x + \beta_{21} z_2 x + \beta_{12} z_1 x^2 + \beta_{22} z_2 x^2 + \varepsilon, \ (6)$$

where

- z_1 is an indicator variable taking the value 1 if enzyme 1 is used and the value 0 otherwise,
- z_2 is an indicator variable taking the value 1 if enzyme 2 is used and the value 0 otherwise, and
- x is the logarithm of the dose in the event an enzyme is used (and can be set to any arbitrary value otherwise).

The two enzyme options and the control treatment in the chicken feed experiment are visualized in Figure 2, along with the corresponding values for the two indicator variables z_1 and z_2 as well as the ranges for the log(dose) of the enzymes.

This model reduces to

$$Y = \beta_0 + \varepsilon$$

when no enzyme is used and $z_1 = z_2 = 0$. Due to the fact that $z_1 = z_2 = 0$, all products $z_1 x, z_2 x, z_1 x^2$ and $z_2 x^2$ are zero regardless of the value of x. Therefore, the value of x is arbitrary in the event no enzyme is used. The model reduces to

$$Y = (\beta_0 + \beta_1) + \beta_{11} x + \beta_{12} x^2 + \varepsilon$$

when enzyme 1 is used and $z_1 = 1$ and $z_2 = 0$, and to

$$Y = (\beta_0 + \beta_2) + \beta_{21}x + \beta_{22}x^2 + \varepsilon$$

when enzyme 2 is used and $z_2 = 1$ and $z_1 = 0$. Splitting up the initial model this way makes clear that it allows for intercepts, main effects and quadratic effects that differ across the enzymes. Thus, we have enzyme-specific intercepts, main effects and quadratic effects. In other words, the intercepts, main effects and quadratic effects are conditional on the level of the branching factor enzyme.

A key aspect of the model for the chicken feed experiment is that, while the branching factor has three levels (no enzyme, enzyme 1 and enzyme 2), there are only two nesting relationships. For this reason, we need two indicator variables z_1 and z_2. The first nested factor, the log(dose) of enzyme 1, enters the model through the cross-products z_1x and z_1x^2. The second nested factor, the log(dose) of enzyme 2, enters the model through the cross-products z_2x and z_2x^2. The nested factor x, the log(dose), does not appear in the model on its own.

4.1.3 Design

If the (log) dose effect in the chicken feed experiment is linear instead of quadratic, then an appropriate model, having five regression coefficients, is

$$Y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \beta_{11} z_1 x + \beta_{21} z_2 x + \varepsilon.$$

When the number of runs is a multiple of 5, the D-optimal design for this model uses the control treatment for one fifth of the runs, the first enzyme for two fifths of the runs, and the second enzyme for the remaining two fifths of the runs. So, the number of runs is not equally divided among the three levels (none, enzyme 1, enzyme 2) of the enzyme factor. Half of the runs with a given enzyme require the lowest dose to be used, while the other half of the runs require the largest dose to be used. So, the D-optimal design involves five test combinations, which should be replicated as evenly as possible. Whenever the number of runs is not a multiple of five, the D-optimality of the

design does not depend on which test combinations are replicated the most often.

If a quadratic dose effect is anticipated, the model in Equation (6), involving seven regression coefficients, is appropriate. D-optimal designs for this model involve seven test combinations, one of which is the control treatment in which no enzyme is used. Three other test combinations use the first enzyme, with its three possible doses. The remaining three test combinations involve the second enzyme, with its three possible doses. Whenever the number of runs is a multiple of 7, each of the 7 test combination has to be used equally often in the D-optimal design. When the number of runs is not a multiple of 7, the numbers of replicates should be as close to being equal as possible.

4.2 An extraction experiment

4.2.1 Problem

A Belgian pharmaceutical company conducted an experiment to optimize the yield and purity of an extraction process. Setting up the experiment and modeling the resulting data was complicated by the fact that the ranges and the effects of two quantitative factors depended on the levels of two categorical factors. This was due to the fact that there were two washing steps in the extraction process, each involving a buffer:

- Two buffer types could be used for the first washing step. The buffer's pH level could range from 6 to 8 for one buffer type, while it was restricted to a range from 7 to 8 for the second buffer type.
- Four buffer types could be used for the second washing step. For buffer type 1, the pH level had to be in the range $[3,4]$. For buffer type 2, the pH level had to be in the range $[3,6]$. For buffer type 3, the pH level had to be in the range $[3.5,5.5]$, and, for buffer type 4, it had to be in the range $[5,6]$.

In this experiment, there are two branching factors (buffer type for washing step 1 and buffer type for washing step 2), two nested factors (the pH level for washing step 1 and the pH level for washing step 2), and six nesting

relationships in total. Two of the nesting relationships occur due to washing step 1, while the other four occur due to washing step 2. As a consequence, there is a nesting relationship and a nested factor for each level of the first branching factor and for each level of the second branching factor. This is different from the proof-of-concept example and from the chicken feed experiment, where there were fewer nesting relationships and nested factors than levels of the branching factor.

This experimental scenario can be viewed as a special case of an experiment with sliding levels (Taguchi; 1987; Hamada and Wu; 1995; Cheng et al.; 2006). The term sliding levels has been used whenever the range of one quantitative factor, the nested factor, depends on the level of another factor, the branching factor. In all published work on the topic of sliding levels, the focus has been on scenarios in which the branching factor was quantitative. In the extraction experiment we consider here, there are two categorical branching factors, each with a nested factor whose range depends on the level of its categorical branching factor. For cases in which both the branching factor and the nested factor are quantitative, we prefer to define the experimental region using inequality constraints (see, for example, Chapter 5 in Goos and Jones (2011)).

4.2.2 Model

Assuming the two washing steps have independent effects on the yield and the purity of the extraction, a sensible model for the data from the extraction experiment is

$$Y = \beta_0 + \beta_{012}z_{12} + \sum_{i=2}^{4}\beta_{02i}z_{2i} + \sum_{i=1}^{2}\beta_{1i}z_{1i}x_{1i} + \sum_{i=1}^{4}\beta_{2i}z_{2i}x_{2i} + \varepsilon, \quad (7)$$

where

- z_{1i} is an indicator variable taking the value 1 if the first washing step uses buffer type *i* and the value 0 otherwise,

- z_{2i} is an indicator variable taking the value 1 if the second washing step uses buffer type i and the value 0 otherwise,

- x_{1i} is the pH level utilized in the first washing step in the event buffer type i is used (and x_{1i} can be set to any arbitrary value in the event buffer type i is not used in washing step 1), and

- x_{2i} is the pH level utilized in the second washing step in the event buffer type i is used (and x_{2i} can be set to any arbitrary value in the event buffer type i is not used in washing step 2).

As there are two possible buffer types for washing step 1 and four possible ones for washing step 2, the model in Equation (7) describes the response in eight different scenarios. The two buffer type options for washing step 1 and the four options for washing step 2 are visualized in Figure 3a and 3b, respectively, along with the corresponding values for the indicator variables and the ranges for the pH level for each choice of buffer type.

In the scenario where both washing steps utilize buffer type 1, $z_{11} = z_{21} = 1,\ z_{12} = z_{22} = z_{23} = z_{24} = 0$, and the model simplifies to

$$Y = \beta_0 + \beta_{11} x_{11} + \beta_{21} x_{21} + \varepsilon.$$

In the scenario where the first washing step uses buffer type 1 and the second washing step uses buffer type i (with $i > 1$), then $z_{11} = z_{2i} = 1$, all remaining indicator variables are zero, and

$$Y = \beta_0 + \beta_{02i} + \beta_{11} x_{11} + \beta_{2i} x_{2i} + \varepsilon.$$

In the scenario where the first washing step uses buffer type 2 and the second washing step uses buffer type 1, then $z_{12} = z_{21} = 1$, all other indicator variables are zero, and

$$Y = \beta_0 + \beta_{012} + \beta_{12} x_{12} + \beta_{21} x_{21} + \varepsilon.$$

Finally, in the scenario where the first washing step uses buffer type 2 and the second washing step uses buffer type i (with $i > 1$), $z_{12} = z_{2i} = 1$, all remaining indicator variables are zero, and

$$Y = \beta_0 + \beta_{012} + \beta_{02i} + \beta_{12}x_{12} + \beta_{2i}x_{2i} + \varepsilon.$$

As a result, the initial model involving four indicator variables allows for buffer-specific intercepts and buffer-specific main effects of the pH level in the two washing steps. Thus, the model involves intercepts and main effects that are conditional on the level of the two categorical branching factors, namely the buffer type in washing step 1 (which has two levels) and the buffer type in washing step 2 (which has four levels). The model in Equation (7) is therefore consonant with Hamada and Wu (1995), who state that the effect of the nested factor is usually modeled separately at each level of the branching factor.

The model we specify here for the extraction experiment is similar to the models used for the proof-of-concept example and the chicken feed experiment in that the six nested factors (two for washing step 1 and four for washing step 2) enter the model through cross-products involving an indicator variable z_{1i} or z_{2i}, depending on whether the nested factor is related to the first or to the second washing step. A difference between the model for the extraction experiment and the previous examples is that not all indicator variables are included in the model separately, to avoid collinearity. More specifically, one indicator variable term per branching factor should be dropped from the model. In our model, we dropped the terms $\beta_{011}z_{11}$ and $\beta_{021}z_{21}$. Which indicator variable term is dropped for each branching factor is arbitrary, except that it impacts the interpretation of β_0. The strategy in which one indicator variable term per branching factor has to be dropped from the model is necessary whenever the number of nesting relationships and nested factors equals the number of levels of the branching factor.

4.2.3 Design

A 24-run D-optimal design for the model in Equation (7) is shown in Table 3. The D-optimal design uses each buffer type for the first washing step equally often, and it also uses each buffer type for the second washing step equally often. For any given buffer type, the low pH level and the high pH level are used the same number of times. Consequently, the D-optimal design exhibits certain kinds of balance in the buffer types and the pH levels. The eight combinations of buffer types from the two washing steps are not equireplicated in the 24-run design. This is different when we increase the number of runs to 32. Here, the D-optimal design has 32 unique runs, each of the eight buffer type combinations appears four times, and a 2^2 factorial design in the pH levels for the two washing steps is used for each combination.

4.3 Magnetic bead experiment

Decrop et al. (2016, 2017) describe a response surface experiment concerning the optical manipulation of magnetic beads. The experiment involved a two-level categorical factor (bead type), a seven-level categorical factor (surfactant) and three quantitative factors (ionic buffer strength, buffer pH and surfactant concentration). The seven levels of the surfactant factor were 'None', 'Brij35', 'Pluronic-F68', 'Tween20', 'Tween40', 'Tween60' and 'Tween80'. Obviously, the factor surfactant concentration is only relevant in the event the level of the surfactant factor differs from 'None'. The investigators expected the effects of the surfactant concentration to depend on the type of surfactant. As they also expected surfactant concentration to be involved in two-way interactions with the factors bead type, ionic buffer strength and buffer pH, this experiment includes both conditional main effects and conditional quadratic effects of the nested factor, surfactant concentration, as well as conditional two-factor interaction effects involving it.

In this experiment, the surfactant factor is a seven-level branching factor, defining six nesting relationships and six nested factors. For all experimental tests using a surfactant, the level of the nested factor, surfactant concentration, was in the range from 0.05% to 1%. The factors bead type,

ionic buffer strength and buffer pH are shared factors, when using the terminology of Hung et al. (2009).

In the supplementary materials, we discuss a suitable model for the magnetic bead experiment. The model involves six indicator variables and 60 regression coefficients. The model reduces to a second-order response surface model in the factors ionic buffer strength and buffer pH for any bead type in the event no surfactant is used. However, whenever a surfactant is used, the model reduces to a second-order response surface model in the factors ionic buffer strength, buffer pH and surfactant concentration for any bead type.

5 Design construction

5.1 Point-exchange algorithm

As can be seen from Chapter 12 in Atkinson et al. (2007), many algorithms for computing optimal experimental designs use a candidate set of test combinations or design points. These algorithms are called point-exchange algorithms, because they iteratively improve the experimental design by exchanging its points with points from the candidate list. Many variations of such candidate-set-based point-exchange algorithms can be found in the literature (see, for example, Fedorov (1972), Cook and Nachtsheim (1980) and Atkinson and Donev (1989)). Atkinson et al. (2007) mention that usually a coarse grid in the experimental region is used as the set of candidate points. However, for problems involving constrained experimental regions, finer grids have substantial added value. Candidate-set-based point-exchange algorithms involve two steps.

First, they create a starting design. It is common to do this by randomly selecting a few points from the candidate set and completing the starting design by sequentially adding the candidate to the design which generates the largest increase in D-optimality criterion value (Galil and Kiefer; 1980; Atkinson and Donev; 1989).

Given a starting design, candidate-set-based point-exchange algorithms proceed sequentially by considering the exchange of each point of the design by a point from the candidate set. The modified Fedorov point-exchange algorithm proposed by Cook and Nachtsheim (1980) does this in a first-improvement fashion, meaning that, as soon as beneficial exchange of a certain design point and a candidate point is discovered, the algorithm proceeds to the next design point and seeks a beneficial exchange for that point. This approach saves computing time when compared to the original Fedorov exchange algorithm, which uses a best-improvement approach. In that approach, every design point is replaced by the candidate that produces the largest increase in D-optimality criterion value, rather than by the first candidate encountered during the search that produces an improved D-optimality criterion. After cycling through the entire design in this manner, the algorithm returns to the first design point and continues to attempt exchanges between the candidate set and the design. This process continues until no more beneficial exchanges can be found during an entire cycle through the design or until a pre-specified number of iterations has taken place.

Because there is no guarantee that a run of the point-exchange algorithm produces a D-optimal design, the algorithm is restarted many times, each time using a new random starting design. The design with the largest D-optimality criterion value encountered during all the iterations is eventually returned.

5.2 Modified coordinate-exchange algorithm

We adapted the coordinate-exchange algorithm of Meyer and Nachtsheim (1995) to deal with the presence of branching factors. As with the original algorithm, our modified coordinate-exchange algorithm involves two stages. First, it creates a feasible starting design. Next, the algorithm improves the starting design matrix coordinate by coordinate (i.e., the algorithm sequentially optimizes the individual settings of the factors, starting with the setting of the first factor in the first run and ending with the setting of the last factor in the last run), where the improvement is in terms of the D-optimality criterion value. Pseudo-code for creating the starting design

and for the improvement stage of our modified coordinate-exchange algorithm appears in the appendix.

The original coordinate-exchange algorithm starts by constructing a completely random design and then sequentially considers each factor setting for each run of the experimental design. If switching a factor setting improves the D-optimality criterion value, the current design is updated with the new factor setting. After proceeding through the entire design run by run and factor setting by factor setting, the algorithm iterates through the entire design over and over until a maximum number of iterations is reached, or until no factor setting is modified in an entire iteration. At that point, one run of the coordinate-exchange algorithm is complete.

Because, as with the point-exchange algorithm, there is no guarantee that a run of the coordinate-exchange algorithm produces a D-optimal design, the coordinate-exchange algorithm is also restarted many times, each time using a new random starting design. The design with the largest D-optimality criterion value encountered during all the iterations is eventually returned.

Our modified coordinate-exchange algorithm follows all these basic steps. Our modification of the algorithm only affects what happens when a branching factor or a nested factor is encountered, both in the construction of the starting design and in the improvement phase.

In both of its steps, our modified coordinate-exchange algorithm distinguishes between branching factors, nested factors and shared factors. To create a starting design, the algorithm randomly selects a level for any branching factor for each run. In the event a level has been selected with which one or more nested factors are associated, a random level is also selected for each of these nested factors.

In the improvement phase, the modified coordinate-exchange algorithm first considers the branching factors and checks whether changing their levels results in a more efficient design. A complication in this process is that changing a branching factor's level to a level with which one or more nested

factors are associated requires considering the levels of these nested factors as well. For simplicity of explanation, we assume that every branching factor has two levels and gives rise to one nesting relationship. Level 0 of a branching factor represents the level corresponding to which there are no nested factors. Level 1 represents the level corresponding to which there is at least one nested factor.

If the current level of a branching factor is zero, then the algorithm switches the level to one. At the same time, the algorithm optimizes the level of all the associated nested factors. If the D-optimality criterion value improves as a result of all these changes, the level of one for the branching factor and the optimized levels of the associated nested factors are retained.

If the current level of a branching factor is one, then the algorithm evaluates the effect of changing the level to zero. If this change improves the D-optimality criterion value, the zero level is retained for the branching factor. Since the nested factors corresponding to the branching factor only appear in the model in product with the branching factor, their levels need not change as the cross-products involving these factors will be zero due to the zero level for the branching factor.

When the modified coordinate-exchange algorithm encounters a nested factor, it optimizes the setting of that factor if its associated branching factor has level 1 and skips the factor when its associated branching factor has level 0.

5.3 Algorithm comparison

Both point-exchange and coordinate-exchange algorithms have strengths and weaknesses. One strength of candidate-set-based point-exchange algorithms is their simple implementation once a candidate set is provided. For problems with few factors, generating an appropriate candidate set is not demanding, even given the special characteristics of designs with nested factors. For example, the chicken feed experiment with quadratic dose effects only requires a candidate set of seven design points: three for the low, high and

middle dose for each of the two enzymes, plus one additional design point corresponding to the control treatment.

Candidate-set-based point-exchange algorithms have weaknesses as well. From the standpoint of the user, the requirement to produce a candidate set, especially for problems with many factors and multiple nesting relationships, may be tedious and error prone. As the number of factors increases, the size of the candidate set grows as the product of the numbers of levels of the factors. This means that the time a candidate-set-based point-exchange algorithm takes to reach completion is exponential in the number of factors. For the small numbers of factors we consider in this paper, computing time issues are not a practical concern. However, if there are many factors each having many levels, computing time issues could dominate other considerations.

While a candidate-set-based point-exchange algorithm needs not be modified to deal with nested factors, we do need to modify the coordinate-exchange algorithm to account for the special requirements posed by nested factors. One consequence of the required modifications is that the term coordinate exchange becomes a bit of a misnomer. This is because, when considering a branching factor, the algorithm may need to change the value of both the branching factor and one or more associated nested factors (and thus multiple coordinates) simultaneously. Our modified coordinate-exchange algorithm therefore bears some similarities with the mixture coordinate-exchange algorithm of Piepel et al. (2005) and the split-plot and split-split-plot coordinate-exchange algorithms of Jones and Goos (2007, 2009), where changing one coordinate also implies changes in at least one other coordinate.

One of the strengths of the coordinate-exchange algorithm is that it does not require a candidate set. It is also unnecessary to discretize the range of continuous factors, since a one-dimensional continuous optimization algorithm such as the one from Brent (1973) can be used to determine optimal values for the levels of the continuous factors. Modern coordinate-exchange

algorithms implementing this approach have been used by Jones and Goos (2012), Ruseckaite et al. (2017) and Huang et al. (2019). The use of a continuous optimizer is especially useful if there are factors that are ingredients of a mixture (Goos et al.; 2016) or if there are linear or nonlinear constraints on the experimental region (Goos and Jones; 2011). The time the coordinate-exchange algorithm takes to reach completion is proportional to the sum of the numbers of levels of the factors rather than the product. This makes the coordinate-exchange computationally feasible even when there are dozens or even hundreds of factors.

For the problems studied here, candidate-set-based point-exchange and coordinate-exchange algorithms found designs with the same D-optimality criterion value, within seconds. However, several years' experience indicates that, for a fixed allotted computed time, the coordinate-exchange algorithm provides more efficient designs than a point-exchange algorithm. This advantage increases with the number of factors considered.

Recently, to avoid the problems inherent to using a candidate set, Huang et al. (2019) proposed a candidate-set-free point-exchange algorithm to optimally design experiments for non-linear models. Given a random starting design, the candidate-set-free algorithm uses a multi-dimensional continuous optimizer, such as the Nelder-Mead or quasi-Newton method, to move current design points to better positions. This avoids the need to discretize the continuous factors. Huang et al. (2019) state that their new algorithm is especially useful for non-linear regression models. This is because, in the absence of constraints on the factor levels, a candidate set with equally spaced levels will permit the construction of a design that is either optimal or very close to optimal for linear models, provided the number of runs is sufficiently large. Modifying the algorithm of Huang et al. (2019) to deal with branching factors and nested factors could be done in a way similar to the way in which we modified the coordinate-exchange algorithm.

6 Screening designs involving conditional main effects

Su and Wu (2017) discuss a novel approach to analyzing data from fractional factorial two-level screening designs. A key aspect of the approach is that it avoids traditional two-factor interaction effects. Instead, it uses conditional main effects. Clearly, this kind of effect is exactly the kind of effect that occurs in the practical examples previously discussed. As Su and Wu (2017) were interested in analysis rather than design, they re-analyzed three data sets published in Montgomery (1991) without discussing how the interest in conditional main effects affects the design of experiments. Therefore, in this section, we create D-optimal designs for two of the examples in Su and Wu (2017), assuming the interest is in main effects of some factors and conditional main effects of other factors.

6.1 Injection molding experiment

The injection molding experiment involved six factors, A,…,F, and used a 2_{IV}^{6-2} design. Su and Wu (2017) identified a model involving a significant conditional main effect for factor A. That main effect occurred at the high level of factor B. For this reason, we calculated a D-optimal design for a model involving (i) the conditional main effect of A for the high level of factor B and (ii) the (unconditional) main effects of factors B,…,F. In this example, the factor B is the branching factor, while the factor A is the nested factor and factors C, D, E and F are shared factors. There is only one nesting relationship. Using the indicator variable approach we outlined in Section 3, the model can be written as

$$Y = \beta_0 + \beta_1 z x_A + \beta_2 z + \beta_3 x_C + \beta_4 x_D + \beta_5 x_E + \beta_6 x_F + \varepsilon,$$

where z is an indicator variable taking the value 1 if factor B acts at its high level and the value 0 otherwise, and x_A, x_C, x_D, x_E and x_F represent the levels of factors A, C, D, E and F. We contrasted the performance of our D-optimal design with that of the 2_{IV}^{6-2} design. The D-optimal design as well as the 2_{IV}^{6-2} design are shown in Table 4.

The D-optimal design in Table 4 involves 4 runs at the low level of factor B and 12 runs at the high level of that factor. The latter runs allow us to quantify the conditional main effect of factor A. In the D-optimal design, there are 6 runs at the high level of factor A and 6 runs at the low level. In the remaining factors, C, D, E and F, the D-optimal design forms an orthogonal design. The standard error for the main effects of these factors is 0.25, whereas the standard errors for the intercept, the main effect of factor B and the conditional main effect of factor A are 0.289. The average variance of the estimates therefore is 0.0714, which corresponds to an A-efficiency of 87.5%. The D-efficiency of the D-optimal design is 92.11%. The column labeled 'D-opt.' in Table 5 provides an overview of these results.

For the model involving the conditional main effect of factor A rather than the usual unconditional main effect, the 2^{6-2}_{IV} design results in standard errors of 0.25 for the intercept and the main effects of the factors B to F, and a standard error of 0.354 for the conditional main effect of factor A. The average variance of the estimates therefore is also 0.0714. The D-efficiency of the 2^{6-2}_{IV} design is 90.57%. So, in this case, the D-optimal design outperforms the standard design in terms of D-efficiency, but not in terms of average variance of the parameter estimates. The results for the 2^{6-2}_{IV} design are shown in the last column of Table 5.

6.2 Aluminum experiment

The aluminum experiment involved six factors (A,…,F) and used a 2^{6-2}_{IV} design as well. Su and Wu (2017) found two active conditional main effects: the factor E had a significant conditional main effect when factor B acted at its high level, and the factor F had a significant conditional main effect when factor A was at its high level. Therefore, we calculated a D-optimal design for a model involving (i) the conditional main effect of E for the high level of factor B, (ii) the conditional main effect of F for the high level of factor A, and (iii) the (unconditional) main effects of factors A,…,D. In this example, the factors A and B are branching factors, while the factors E and F are nested factors and the factors C and D are shared factors. There are two nesting relationships.

Using the indicator variable approach we outlined in Section 3, the model for the aluminum experiment can be written as

$$Y = \beta_0 + \beta_1 z_A + \beta_2 z_B + \beta_3 x_C + \beta_4 x_D + \beta_5 z_B x_E + \beta_6 z_A x_F + \varepsilon,$$

where z_A is an indicator variable taking the value 1 if factor A acts at its high level and the value 0 otherwise, z_B is an indicator variable taking the value 1 if factor B acts at its high level and the value 0 otherwise, and x_C, x_D, x_E, and x_F represent the levels of factors C, D, E and F. We contrasted the performance of our D-optimal design for this model with that of the 2^{6-2}_{IV} design. The D-optimal design as well as the 2^{6-2}_{IV} design are shown in Table 6.

The D-optimal design in Table 6 involves 4 runs at the low level of factor A (and 12 runs at its high level) as well as 6 runs at the low level of B (and 10 runs at its high level). Consequently, 12 of the 16 runs allow the conditional main effect of factor F to be studied and 10 of the 16 runs provide information concerning the conditional main effect of factor E. The standard errors and the D- and A-efficiencies are listed in Table 7, along with the corresponding values produced by the 2^{6-2}_{IV} design. The table shows that the conditional main effects of the factors E and F are estimated more precisely from the D-optimal design than from the standard design, and that the main effects of the factors A and B are estimated more precisely using the standard design. On average, however, the variance of the parameter estimates is smaller for the D-optimal design than for the standard design. This results in a better A-efficiency for the D-optimal design than for the standard design. Also, the D-efficiency of the D-optimal design is better than that of the 2^{6-2}_{IV} design.

7 Discussion

This paper provides a method for incorporating nested factors in designed experiments. These are factors that are meaningful only for certain levels of another factor or have ranges and effects that depend on the level of another factor. We call the factor within which other factors are nested branching factors. In this work, we discuss the modeling of data from experiments

involving these kinds of factors as well as the optimal design of such experiments.

We applied our optimal design approach to a proof-of-concept example and to two examples in Su and Wu (2017), and we discuss three recent experiments in bioscience engineering. The examples in Su and Wu (2017) involved a re-analysis of several experimental data sets where conditional main effects turned out to be useful descriptions. Our optimal designs for two of these examples involve pretending that the conditional main effects that Su and Wu (2017) found were a result of the structure of the process and could be anticipated. For their examples, of course, this is not the case. Our point was to show that, if one did have a system that allowed for identifying such possibilities, an optimal design with the same number of runs can yield more efficient parameter estimates.

We have given several examples of scenarios we have encountered in practice. This was to try to help practitioners to identify these kinds of problems, and to demonstrate how indicator variables can be used to build suitable regression models. One of our examples involved 'sliding levels', which are ranges of factors that depend on the levels of other factors. The resulting design can be viewed as a nested design (Montgomery; 1991), but with fixed factor levels rather than randomly selected factor levels (Hamada and Wu; 1995).

Our algorithm for finding optimal designs for models with conditional effects is available in JMP Scripting Language. The Custom Design tool in JMP can also generate these designs through a clever use of its ability to specify disallowed factor level combinations.

Further research in this direction would be to find optimal designs for mixture experiments where the number of ingredients in each formulation is limited to a proper subset of the total number of potential ingredients. Another useful extension of our work would be to account for heterogeneous variances in the responses. This is because different levels of a branching factor may give rise

to unequal variances. For instance, the proof-of-concept example in Section 2 compares an old and a new machine. It is conceivable that the run-to-run variation for the new machine is substantially smaller than that for the old machine. This can be taken into account by generating optimal designs based on the weighted least squares estimator instead of the ordinary least squares one.

Supplementary materials

- **Magnetic bead experiment**: Full description of the magnetic bead experiment, including the model and the definitions of the indicator variables.
- **Illustration of the modified coordinate-exchange algorithm**: A detailed description of how the modified coordinate-exchange algorithm works when optimizing a 4-run experimental design for the proof-of-concept example in Section 2.
- **JMP scripts**: JMP scripts with implementations of the modified coordinate-exchange algorithm for the chicken feed, injection molding and aluminum examples in the paper.

Acknowledgment

Appendix: Pseudo-Code of the Modified Coordinate-Exchange Algorithm

Our coordinate-exchange algorithm involves two stages. First, it creates a random experimental design with the desired structure. Next, it iteratively improves the starting design. This improvement is performed coordinate by coordinate for shared factors, but, for branching factors and nested factors, it is performed at the same time. To optimize the readability of the pseudo-code in this appendix, we assume that every branching factor has two levels and

gives rise to one nesting relationship. Level 0 of a branching factor represents the level corresponding to which there are no nested factors. Level 1 represents the level corresponding to which there is at least one nested factor. We also assume that the nested factors and the shared factors are all quantitative, that they can take any level on the $[-1,+1]$ interval and that there are no constraints on the levels of the factors. We denote the number of experimental runs by n and the total number of factors by m, and we let 'CurrentDet' represent the D-optimality criterion value of the best experimental design encountered in the course of one run of our modified coordinate-exchange algorithm. Finally, the improvement phase of the algorithm involves a boolean variable, named 'Improvement', which signals whether an improvement of the design has been achieved during one pass through the entire design. That variable can take two values: 'Yes' or 'No'. At the start of the improvement phase, this boolean variable is initialized to the value 'No'.

Constructing a starting design

1. For $i = 1, \ldots, n$ do

   (a) For $j = 1, \ldots, m$ do

      i. If factor j is a branching factor then set its level for run i to 0 or 1 at random

      ii. Else set its level for run i to a randomly chosen value on the $[-1,+1]$ interval

2. CurrentDet ← D-criterion value of starting design

3. If CurrentDet = 0, return to Step 1.

Improvement phase

1. For $i = 1, \ldots, n$ do

   (a) For $j = 1, \ldots, m$ do

      i. If factor j is a branching factor then

         • If factor j's current level is 0 at run i then

– Set factor j's level at run i to 1

– Determine the optimal level(s) for the corresponding nested factor(s) at run i

– NewDet $\leftarrow$ D-criterion value of the resulting design

– If NewDet > CurrentDet

    * CurrentDet $\leftarrow$ NewDet

    * Improvement $\leftarrow$ 'Yes'

    * Set the nested factor(s) to their optimal level(s) at run i

– Else set factor j's level at run i back to 0

- Else if factor j's current level is 1 at run i then

– Set factor j's level at run i to 0

– NewDet $\leftarrow$ D-criterion value of the resulting design

– If NewDet > CurrentDet

    * CurrentDet $\leftarrow$ NewDet

    * Improvement $\leftarrow$ 'Yes'

– Else set factor j's level at run i back to 1

ii. Else if factor j is a shared factor or a nested factor the associated branching factor of which has level 1

- Determine the optimal level for factor j at run i

- NewDet ← D-criterion value of the resulting design

- If NewDet > CurrentDet

    – CurrentDet ← NewDet

    – Improvement ← 'Yes'

    – Set factor j to its optimal level at run i

  iii. Else if factor j is a nested factor the associated branching factor of which has level 0, go to the next factor

2. If Improvement = 'Yes' then

    (a) Improvement ← 'No'

    (b) Return to Step 1 of the improvement phase

3. Else report the current design and the value of CurrentDet

References

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.

Atkinson, A. C., Donev, A. N. and Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS*, Oxford: Oxford University Press.

Brent, R. P. (1973). *Algorithms for Minimization without Derivatives*, Englewood Cliffs, NJ: Prentice Hall.

Cheng, S.-W., Wu, C. F. J. and Huwang, L. (2006). Statistical modeling for experiments with sliding levels, *Lecture Notes-Monograph Series* **52**: 245–256.

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

Decrop, D., Brans, T., Gijsenbergh, P., Lu, J., Spasic, D., Kokalj, T., Beunis, F., Goos, P., Puers, R. and Lammertyn, J. (2016). Optical manipulation of single magnetic beads in a microwell array on a digital microfluidic chip, *Analytical Chemistry* **88**(17): 8596–8603.

Decrop, D., Lammertyn, J. and Goos, P. (2017). Optimal experimental design for efficient optical manipulation of magnetic beads seeded in a microwell array, *Journal of Quality Technology* **49**(4): 402–417.

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

Galil, Z. and Kiefer, J. (1980). Time- and space-saving computer methods, related to Mitchell's DETMAX, for finding D-optimum designs, *Technometrics* **21**: 301–313.

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

Goos, P., Jones, B. and Syafitri, U. (2016). I-optimal design of mixture experiments, *Journal of the American Statistical Association* **111**(514): 899–911.

Hamada, M. and Wu, C. F. J. (1995). The treatment of related experimental factors by sliding levels, *Journal of Quality Technology* **27**(1): 45–55.

Huang, Y., Gilmour, S. G., Mylona, K. and Goos, P. (2019). Optimal design of experiments for non-linear response surface models, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **68**: To appear.

Hung, Y., Joseph, V. R. and Melkote, S. N. (2009). Design and analysis of computer experiments with branching and nested factors, *Technometrics* **51**: 354–365.

Jones, B. and Goos, P. (2007). A candidate-set-free algorithm for generating D-optimal split-plot designs, *Journal of the Royal Statistical Society. Series C* **56**: 347–364.

Jones, B. and Goos, P. (2009). D-optimal design of split-split-plot experiments, *Biometrika* **96**: 67–82.

Jones, B. and Goos, P. (2012). An algorithm for finding D-efficient equivalent-estimation second-order split-plot designs, *Journal of Quality Technology* **44**(4): 363–374.

Mak, S. and Wu, C. F. J. (2018). cmenet: a new method for bi-level variable selection of conditional main effects, *Journal of the American Statistical Association*. To appear.

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

Montgomery, D. C. (1991). *Design and Analysis of Experiments*, New York: Wiley.

Palmers, S., Grauwet, T., Buvé, C., Vanratingen, K., Kebede, B., Goos, P., Hendrickx, M. and Van Loey, A. (2016). Relative importance and interactions of furan precursors in sterilised, vegetable-based food systems, *Food Additives and Contaminants Part A* **33**(2): 193–206.

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**(4): 579–594.

Ruseckaite, A., Goos, P. and Fok, D. (2017). Bayesian D-optimal choice designs for mixtures, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **66**(2): 363–386.

Schoen, E. D., Jones, B. and Goos, P. (2011). A split-plot experiment with factor-dependent whole-plot sizes, *Journal of Quality Technology* **43**: 66–79.

Su, H. and Wu, C. F. J. (2017). CME analysis: A new method for unraveling aliased effects in two-level fractional factorial experiments, *Journal of Quality Technology* **49**(1): 1–10.

Taguchi, G. (1987). *System of Experimental Design*, Unipub/Kraus International Publications, White Plains, NY.

Wu, C. F. J. (2015). Post-Fisherian experimentation: From physical to virtual, *Journal of the American Statistical Association* **110**(510): 612–620.
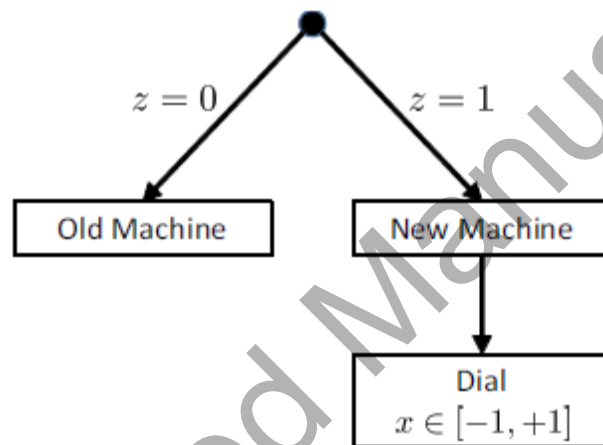
Fig. 1 Diagram showing the two machine options in the proof-of-concept example, the dial factor x for the new machine and the indicator variable z.
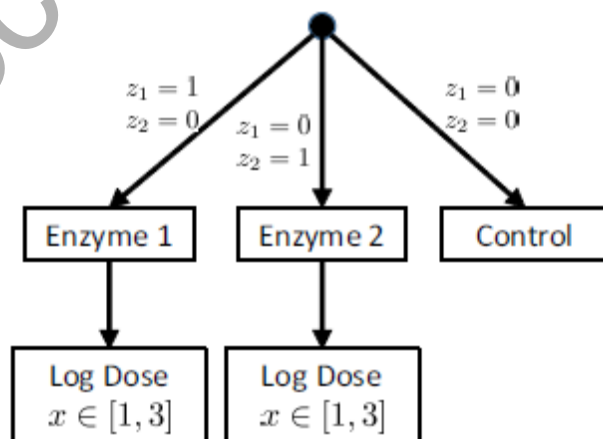
Fig. 2 Diagram showing the two enzyme options and the control treatment and the indicator variables required for the chicken feed experiment.
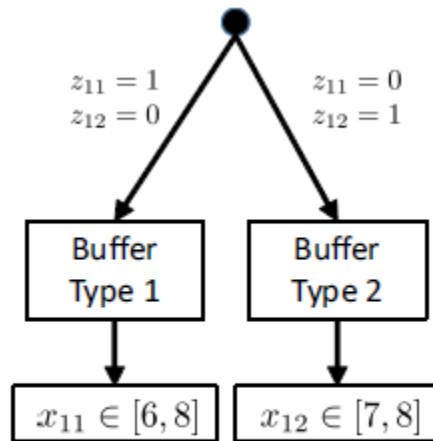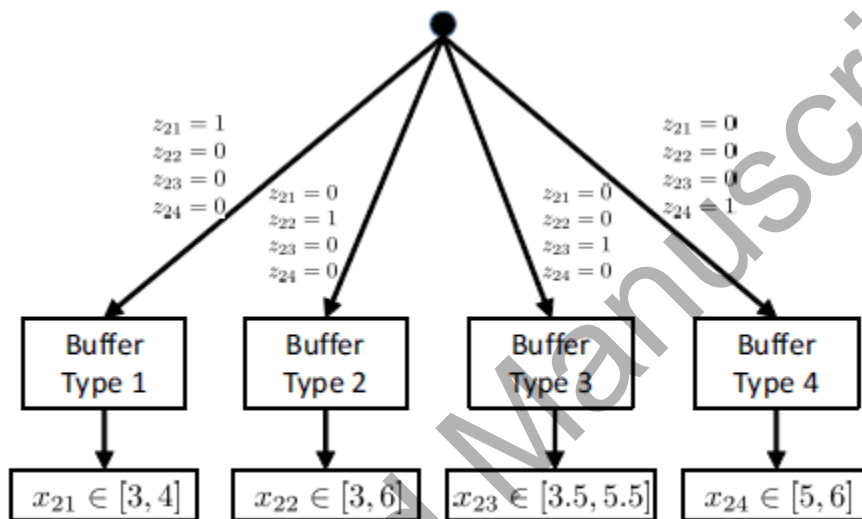
(a) Washing step 1

(b) Washing step 2

Fig. 3 Diagrams showing the buffer type options, the indicator variables required and the ranges for the pH level for each choice of buffer type.

Table 1 Possible design strategy to deal with the two-factor example in Section 2.

Run	Machine	Dial	z	x
1	Old	N.A.	0	N.A.
2	Old	N.A.	0	N.A.

Run	Machine	Dial	z	x
⋮	⋮	⋮	⋮	⋮
$n/2$	Old	N.A.	0	N.A.
$n/2 + 1$	New	Low	1	−1
⋮	⋮	⋮	⋮	⋮
$3n/4$	New	Low	1	−1
$3n/4 + 1$	New	High	1	+ 1
⋮	⋮	⋮	⋮	⋮
n	New	High	1	+ 1

Table 2 Alternative designs, standard errors of estimates as well as efficiencies for the two-factor proof-of-concept example in Section 2, assuming the model in Equation (1) is used.

			$r \times 2^2$			D-optimal			
			8 runs	12 runs	16 runs	8 runs	8 runs	12 runs	16 runs	16 runs
	Machine	Dial	$r = 2$	$r = 3$	$r = 4$	(I)	(II)		(I)	(II)
Replicates	Old	N.A.	4	6	8	3	2	4	5	6
	New	Low	2	3	4	3	3	4	6	5
	New	High	2	3	4	2	3	4	5	5
Diagnostics	s.e.(β_0)		0.354	0.289	0.250	0.368	0.408	0.306	0.270	0.258
	s.e.(β_1)		0.354	0.289	0.250	0.368	0.408	0.306	0.270	0.258
	s.e.(β_2)		0.500	0.408	0.354	0.456	0.408	0.354	0.303	0.316
	D-eff.		79.37	79.37	79.37	82.55	82.55	83.99	83.68	83.68
	A-eff.		75.00	75.00	75.00	78.26	75.00	80.00	78.95	80.36

Table 3 24-run D-optimal design for the extraction experiment in Section 4.2.

Run	Buffer Type		Uncoded pH level		Coded pH level	
	Step 1	Step 2	Step 1	Step 2	Step1	Step 2
1	1	1	6	3	−1	−1
2	1	1	6	4	−1	+1
3	1	1	8	3	+1	−1
4	1	1	8	4	+1	+1
5	1	2	6	6	−1	+1
6	1	2	8	3	+1	−1
7	1	3	6	5.5	−1	+1
8	1	3	8	3.5	+1	−1
9	1	3	8	5.5	+1	+1
10	1	4	6	5	−1	−1
11	1	4	6	6	−1	+1
12	1	4	8	5	+1	−1
13	2	1	7	4	−1	+1
14	2	1	8	3	+1	−1
15	2	2	7	3	−1	−1
16	2	2	7	6	−1	+1
17	2	2	8	3	+1	−1
18	2	2	8	6	+1	+1
19	2	3	7	3.5	−1	−1
20	2	3	7	5.5	−1	+1
21	2	3	8	3.5	+1	−1
22	2	4	7	6	−1	+1
23	2	4	8	5	+1	−1
24	2	4	8	6	+1	+1

Table 4 D-optimal design and 2^{6-2}_{IV} design for the injection molding experiment.

Run	D-optimal design						2^{6-2}_{IV} design					
	A	B	C	D	E	F	A	B	C	D	E	F
1	N.A.	−1	−1	−1	−1	−1	N.A.	−1	−1	−1	−1	−1
2	N.A.	−1	−1	−1	+1	+1	N.A.	−1	−1	−1	+1	−1
3	N.A.	−1	+1	+1	−1	−1	N.A.	−1	−1	+1	−1	+1
4	N.A.	−1	+1	+1	+1	+1	N.A.	−1	−1	+1	+1	+1
5	−1	+1	−1	−1	+1	+1	N.A.	−1	+1	−1	−1	+1
6	−1	+1	−1	+1	−1	−1	N.A.	−1	+1	−1	+1	+1
7	−1	+1	−1	+1	+1	−1	N.A.	−1	+1	+1	−1	−1
8	−1	+1	+1	−1	−1	−1	N.A.	−1	+1	+1	+1	−1
9	−1	+1	+1	−1	−1	+1	−1	+1	−1	−1	+1	+1
10	−1	+1	+1	+1	+1	+1	−1	+1	−1	+1	+1	−1
11	+1	+1	−1	−1	+1	−1	+1	+1	−1	−1	−1	+1
12	+1	+1	−1	+1	−1	+1	+1	+1	+1	−1	+1	+1
13	+1	+1	−1	+1	−1	+1	+1	−1	−1	−1	−1	+1
14	+1	+1	+1	−1	−1	+1	+1	−1	+1	−1	−1	−1
15	+1	+1	+1	−1	+1	−1	+1	+1	−1	+1	−1	−1
16	+1	+1	+1	+1	+1	−1	+1	+1	+1	+1	+1	+1

Table 5 Standard errors as well as D- and A-efficiencies for the D-optimal design and the 2^{6-2}_{IV} design for the injection molding experiment.

Effect	D-opt.	2^{6-2}_{IV}
Intercept	0.289	0.25
A|B+	0.289	0.354
B	0.289	0.25

Effect	D-opt.	2_{IV}^{6-2}
C	0.25	0.25
D	0.25	0.25
E	0.25	0.25
F	0.25	0.25
D-eff.	92.11%	90.57%
A-eff.	87.50%	87.50%

Table 6 D-optimal design and 2_{IV}^{6-2} design for the aluminum experiment.

Run	D-optimal design						2_{IV}^{6-2} design					
	A	B	C	D	E	F	A	B	C	D	E	F
1	−1	+1	+1	+1	+1	N.A.	−1	−1	−1	−1	N.A.	NA
2	−1	−1	−1	−1	N.A.	N.A.	−1	−1	−1	+1	N.A.	NA
3	−1	+1	−1	+1	−1	N.A.	−1	−1	+1	−1	N.A.	NA
4	−1	−1	+1	−1	N.A.	N.A.	−1	−1	+1	+1	N.A.	NA
5	+1	+1	+1	−1	−1	+1	−1	+1	−1	−1	+1	NA
6	+1	+1	−1	−1	+1	+1	−1	+1	−1	+1	+1	NA
7	+1	+1	+1	−1	+1	−1	−1	+1	+1	−1	−1	NA
8	+1	+1	−1	−1	−1	−1	−1	+1	+1	+1	−1	NA
9	+1	−1	−1	+1	N.A.	−1	+1	−1	−1	−1	N.A.	−1
10	+1	+1	+1	+1	−1	+1	+1	−1	−1	+1	N.A.	+1
11	+1	−1	+1	−1	N.A.	+1	+1	−1	+1	−1	N.A.	+1
12	+1	−1	+1	+1	N.A.	−1	+1	−1	+1	+1	N.A.	−1
13	+1	+1	−1	−1	−1	−1	+1	+1	−1	−1	−1	+1
14	+1	−1	−1	+1	N.A.	+1	+1	+1	−1	+1	−1	−1
15	+1	+1	+1	+1	+1	−1	+1	+1	+1	−1	+1	−1
16	+1	+1	−1	+1	+1	+1	+1	+1	+1	+1	+1	+1

Run   D-optimal design 2^{6-2}_{IV} design

Table 7 Standard errors as well as D- and A-efficiencies for the D-optimal design and the 2^{6-2}_{IV} design for the aluminum experiment.

Effect	D-opt.	2^{6-2}_{IV}
Intercept	0.292	0.25
A	0.292	0.25
B	0.261	0.25
C	0.253	0.25
D	0.253	0.25
E|B+	0.324	0.354
F|A+	0.289	0.354
D-eff.	84.44%	82.03%
A-eff.	78.75%	77.78%