Computational Statistics and Data Analysis 54 (2010) 3381–3391

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

Update formulas for split-plot and block designs

Heidi Arnouts *, Peter Goos

Universiteit Antwerpen, Faculty of Applied Economics, Prinsstraat 13, 2000 Antwerpen, Belgium

A R T I C L E   I N F O

Article history:
Received 10 December 2008
Received in revised form 13 November 2009
Accepted 14 November 2009
Available online 29 November 2009

A B S T R A C T

For the algorithmic construction of optimal experimental designs, it is important to be able to evaluate small modifications of given designs in terms of the optimality criteria at a low computational cost. This can be achieved by using powerful update formulas for the optimality criteria during the design construction. The derivation of such update formulas for evaluating the impact of changes to the levels of easy-to-change factors and hard-to-change factors in split-plot designs as well as the impact of a swap of points between blocks or whole plots in block designs or split-plot designs is described.

1. Introduction

Update formulas for the inverse of the information matrix in a regression analysis after deleting or adding one or more points have been known for many years. They have, for instance, proven useful for evaluating the impact of dropping outliers or influential points on the regression. The same and other update formulas have also been used extensively in the construction of experimental designs (see, for instance, Eccleston (1980), John and Whitaker (2000), John and Williams (2000), John (2001), Nguyen (1994), Nguyen and Liu (2008), and Nguyen and Williams (2006)). Update formulas for the determinant of the information matrix have played an important role in the development of algorithms for constructing D-optimal designs. For instance, the K-exchange algorithm by Johnson and Nachtsheim (1983) was justified by the update formula for deleting a point from a design, and the KL-exchange algorithm by Atkinson and Donev (1989) was essentially motivated by an update formula for the exchange of a design point with a candidate point (for a detailed discussion of these algorithms, see Nguyen and Miller (1992)). The coordinate-exchange algorithm for D-, A- and V-optimal designs introduced by Meyer and Nachtsheim (1995) also heavily uses computationally cheap update formulas for evaluating the design criterion after changes to a design.

In recent years much work has been done on the development of algorithms to compute optimal split-plot and blocked response surface designs. Goos and Vandebroek (2003) used a point-exchange algorithm to compute a D-optimal split-plot design for given numbers and sizes of whole plots. Jones and Goos (2007) did the same making use of a coordinate-exchange algorithm. In these papers the whole-plot size is equal for every whole-plot whereas in Goos and Vandebroek (2001b, 2004) exchange algorithms were developed to construct D-optimal split-plot designs where this constraint is relaxed. A point-exchange algorithm was also implemented in Goos and Vandebroek (2001a) to find D-optimal response surface designs for blocked experiments with equal block sizes and random block effects. In Kessels et al. (2008) a point-exchange algorithm was developed to solve a conjoint design problem, based on the optimal design approach for blocked experiments with heterogeneous block sizes. It is clear that two types of algorithms have been used to search for optimal split-plot and block designs, namely point-exchange and coordinate-exchange algorithms. Point-exchange algorithms make use of a set of candidate design points that cover the experimental region well. At each step of the algorithm, an entire design point

* Corresponding author.
E-mail address: heidi.arnouts@ua.ac.be (H. Arnouts).

0167-9473/$ – see front matter
doi:10.1016/j.csda.2009.11.014

is replaced by a point from the candidate set and this change is evaluated in terms of the optimality criterion. Coordinate-exchange algorithms do not require the construction of a set of candidate points. At each step of these algorithms, only the level of a single factor is modified rather than a complete design point.

The published papers on optimal split-plot and blocked response surface designs concentrate on the construction of D-optimal designs. This means that the computation of the determinant of the information matrix, which is of order n^3 for a matrix of rank n, absorbs a considerable fraction of the run-time of the design construction algorithms. In order to find good designs within a reasonable amount of time, it is therefore crucial to make use of update formulas that involve the computation of determinants of low-rank matrices only.

In addition, it is important to have fast update procedures for the inverse of the information matrix because the update formulas for its determinant depend on it. A fast update of the inverse of the information matrix also opens the prospect of computing A- and V-optimal split-plot and block designs at a lower computational cost.

In this paper update formulas for the determinant and the inverse of the information matrix of split-plot and block designs with random block effects are presented. These speed up the construction of D-, A- or V-optimal split-plot and block designs. The update formulas are based on the random intercept model typically used to analyze data from split-plot and block designs. The essential difference between blocked and split-plot designs is that split-plot designs are used in cases where the levels of some factors, the so-called hard-to-change factors, are difficult to reset. The levels of these factors are therefore held constant for several successive runs, leading to blocks of runs at one level for each of the hard-to-change factors. In the split-plot literature, these blocks are called whole plots. The remaining factors in a split-plot experiments are reset independently for each run and called easy-to-change factors. As opposed to split-plot designs, block designs only involve easy-to-change factors. The runs in a block design are grouped in blocks because they cannot be carried out under homogeneous conditions. In this article, the update formulas are described for the more complicated case of split-plot designs. Each of the update formulas for easy-to-change factors for split-plot designs can be adapted easily for the construction of block designs. This is explained in the discussion.

2. Statistical model and algebraic formulas

2.1. Statistical model

The statistical model corresponding to a design with b whole plots of sizes $k_1, \ldots, k_b$ can be written as

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{Y}$ is an n-dimensional response vector, $\mathbf{X}$ represents the $n \times p$ design matrix which contains the settings of the various experimental factors and their model expansion, $\boldsymbol{\beta}$ is the p-dimensional vector of the factor effects and $\mathbf{Z}$ is an $n \times b$ matrix of zeroes and ones, with the (j, i)th element equal to one if the jth run belongs to the ith whole plot, and zero otherwise. Finally, the b-dimensional vector $\boldsymbol{\gamma}$ and the n-dimensional vector $\boldsymbol{\varepsilon}$ contain the whole-plot effects and the random errors, respectively. It is assumed that

$$\mathrm{E}(\boldsymbol{\varepsilon}) = \mathbf{0}_n \quad \text{and} \quad \mathrm{cov}(\boldsymbol{\varepsilon}) = \sigma_\varepsilon^2 \mathbf{I}_n, \tag{2}$$

$$\mathrm{E}(\boldsymbol{\gamma}) = \mathbf{0}_b \quad \text{and} \quad \mathrm{cov}(\boldsymbol{\gamma}) = \sigma_\gamma^2 \mathbf{I}_b, \tag{3}$$

and

$$\mathrm{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\gamma}) = \mathbf{0}_{b \times n}. \tag{4}$$

When using this model, the covariance matrix of the responses is

$$\mathbf{V} = \sigma_\varepsilon^2 \mathbf{I}_n + \sigma_\gamma^2 \mathbf{Z}\mathbf{Z}', \tag{5}$$

where the variances σ_ε^2 and σ_γ^2 are called the error variance and whole-plot variance, respectively. This matrix can be written as

$$\mathbf{V} = \mathrm{diag}(\mathbf{V}_1, \ldots, \mathbf{V}_b), \tag{6}$$

with compound symmetric

$$\mathbf{V}_i = \sigma_\varepsilon^2 \mathbf{I}_{k_i} + \sigma_\gamma^2 \mathbf{1}_{k_i} \mathbf{1}_{k_i}', \tag{7}$$

when the runs of the experiment are arranged per whole plot. In this expression, $\mathbf{I}_{k_i}$ is the k_i-dimensional identity matrix and $\mathbf{1}_{k_i}$ is a k_i-dimensional vector of ones.

The unknown model parameter $\boldsymbol{\beta}$ in (1) can be estimated using the generalized least squares estimator

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{V}^{-1}\boldsymbol{Y}. \tag{8}$$

The covariance matrix of this estimator equals

$$\mathrm{var}(\hat{\boldsymbol{\beta}}) = \left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-1}, \tag{9}$$

and the information matrix on the unknown parameters $\boldsymbol{\beta}$ is given by

$$\mathbf{M} = \mathbf{X}'\mathbf{V}^{-1}\mathbf{X}. \tag{10}$$

The inverse covariance matrix of the vector of responses can be written as

$$\mathbf{V}^{-1} = \sigma_\varepsilon^{-2}\mathbf{I}_n - \sigma_\varepsilon^{-2}\eta\mathbf{Z}(\mathbf{I}_b + \eta\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}', \tag{11}$$

where $\eta = \sigma_\gamma^2/\sigma_\varepsilon^2$ measures the extent to which observations are correlated. When ordering the runs per whole plot, Eq. (11) can be written as

$$\mathbf{V}^{-1} = \mathrm{diag}(\mathbf{V}_1^{-1}, \ldots, \mathbf{V}_b^{-1}), \tag{12}$$

with

$$\mathbf{V}_i^{-1} = \sigma_\varepsilon^{-2}\mathbf{I}_{k_i} - \sigma_\varepsilon^{-2}\frac{\eta}{1+\eta k_i}\mathbf{1}_{k_i}\mathbf{1}'_{k_i}. \tag{13}$$

This makes it possible to rewrite the information matrix (10) as

$$\begin{aligned}\mathbf{M} &= \sigma_\varepsilon^{-2}\mathbf{X}'\mathbf{X} - \sigma_\varepsilon^{-2}\eta\mathbf{X}'\mathbf{Z}(\mathbf{I}_b + \eta\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{X} \\ &= \sigma_\varepsilon^{-2}\mathbf{X}'\mathbf{X} - \sigma_\varepsilon^{-2}\sum_{i=1}^{b}\left[\frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)'\right] \\ &= \sigma_\varepsilon^{-2}\left[\sum_{i=1}^{b}\sum_{j=1}^{k_i}\boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)\boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)\right] - \sigma_\varepsilon^{-2}\sum_{i=1}^{b}\left[\frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)'\right],\end{aligned} \tag{14}$$

where $\mathbf{X}_i$ is the part of the design matrix $\mathbf{X}$ corresponding to the ith whole plot and $\boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)$ is the model expansion of the jth design point in the ith whole plot. The vector $\boldsymbol{w}_i$ denotes the levels of the hard-to-change factors within whole plot i. The vector $\boldsymbol{t}_{ij}$ denotes the levels of the easy-to-change factors at the jth run within the ith whole plot. Without loss of generality it is assumed in the following sections that $\sigma_\varepsilon^2 = 1$.

The D-, A- and V-optimality criteria are direct functions of the information matrix $\mathbf{M}$. A D-optimal design maximizes $|\mathbf{M}|$, whereas an A-optimal design minimizes $\mathrm{tr}\left(\mathbf{M}^{-1}\right)$. A V-optimal design minimizes

$$\int_{(\boldsymbol{w},\boldsymbol{t})\in\chi} \boldsymbol{f}'\left(\boldsymbol{w}, \boldsymbol{t}\right)\mathbf{M}^{-1}\boldsymbol{f}\left(\boldsymbol{w}, \boldsymbol{t}\right)\mathrm{d}\boldsymbol{w}\mathrm{d}\boldsymbol{t},$$

where χ represents the region of interest for $\boldsymbol{w}$ and $\boldsymbol{t}$. As shown by Meyer and Nachtsheim (1995), this can be rewritten as $\mathrm{tr}\left(\mathbf{M}^{-1}\mathbf{B}\right)$, where

$$\mathbf{B} = \int_{(\boldsymbol{w},\boldsymbol{t})\in\chi} \boldsymbol{f}\left(\boldsymbol{w}, \boldsymbol{t}\right)\boldsymbol{f}'\left(\boldsymbol{w}, \boldsymbol{t}\right)\mathrm{d}\boldsymbol{w}\mathrm{d}\boldsymbol{t}.$$

The key role $|\mathbf{M}|$ and $\mathbf{M}^{-1}$ play in these optimality criteria means that fast update formulas for them are desirable for the algorithmic search for optimal designs.

In the remainder of this paper a series of different possible scenarios and the corresponding update formulas are presented. These scenarios are applicable to the situation where the number of runs is fixed for every whole plot as well as to the situation where each k_i is free to be chosen.

2.2. Algebraic formulas for the derivation of the update formulas

The update formulas derived in this paper make use of two algebraic formulas, proofs of which can be found in Harville (1997), for matrices of the form $\mathbf{R} + \mathbf{STU}$. In this expression, $\mathbf{R}$ and $\mathbf{T}$ are nonsingular $r \times r$ and $t \times t$ matrices, respectively, whereas $\mathbf{S}$ and $\mathbf{U}$ are $r \times t$ and $t \times r$ matrices, respectively. The formulas are given by

$$\begin{aligned}|\mathbf{R} + \mathbf{STU}| &= |\mathbf{R}|\,|\mathbf{T}|\,\left|\mathbf{T}^{-1} + \mathbf{UR}^{-1}\mathbf{S}\right| \\ &= |\mathbf{R}|\,\left|\mathbf{I} + \mathbf{TUR}^{-1}\mathbf{S}\right|\end{aligned} \tag{15}$$

and

$$(\mathbf{R} + \mathbf{STU})^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{S}\left(\mathbf{T}^{-1} + \mathbf{UR}^{-1}\mathbf{S}\right)^{-1}\mathbf{UR}^{-1}. \tag{16}$$

The latter expression is sometimes referred to as the Sherman–Morrison–Woodbury formula. It is shown below that the information matrix in (14) after a change to one or more levels of easy-to-change or hard-to-change factors, or after a swap of two design points in different whole plots, in general can be expressed as

$$\mathbf{M}^* = \mathbf{M} + \mathbf{U}'\mathbf{D}\mathbf{U}, \tag{17}$$

where $\mathbf{M}$ and $\mathbf{M}^*$ are the information matrices before and after the change, respectively, $\mathbf{D}$ is a d-dimensional diagonal matrix and $\mathbf{U}$ is a $d \times p$ matrix. The update of the determinant of this information matrix can be done using a formula of the form

$$\left|\mathbf{M}^*\right| = |\mathbf{M}|\,|\mathbf{D}|\,\left|\mathbf{D}^{-1} + \mathbf{U}\mathbf{M}^{-1}\mathbf{U}'\right|, \tag{18}$$

whereas the inverse of the information matrix can be updated as follows:

$$\mathbf{M}^{*-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}'\left(\mathbf{D}^{-1} + \mathbf{U}\mathbf{M}^{-1}\mathbf{U}'\right)^{-1}\mathbf{U}\mathbf{M}^{-1}. \tag{19}$$

The update formulas are extremely useful because $|\mathbf{M}|$ and $\mathbf{M}^{-1}$ are being stored during the operation of the algorithm, and $|\mathbf{D}|$ as well as $\mathbf{D}^{-1}$ are easy to compute as $\mathbf{D}$ is a diagonal matrix. The actual update formulas are detailed in the following sections.

3. Changes to easy-to-change factor levels

Changes to the level of one easy-to-change factor for one of the runs occur in both coordinate-exchange and point-exchange algorithms, whereas changes to the level of more than one such factor are made in point-exchange algorithms only. When an adjustment is made to the easy-to-change factors $\boldsymbol{t}_{ij}$ of run j in whole plot i, only the corresponding row in the design matrix $\mathbf{X}$ is affected. Using $\boldsymbol{f}'(\boldsymbol{w}_i, \boldsymbol{t}_{ij})$ to denote the original of the affected row and $\boldsymbol{f}'(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*)$ for the adjustment, the information matrix (14) can be updated as follows:

$$\mathbf{M}^* = \mathbf{M} - \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)\boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) + \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right)\boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right) + \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' - \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)', \tag{20}$$

where $\mathbf{M}^*$ and $\mathbf{X}_i^*$ represent the updated versions of $\mathbf{M}$ and $\mathbf{X}_i$, respectively. Moreover the term $\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}$ can easily be computed as

$$\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i} = \mathbf{X}_i'\mathbf{1}_{k_i} - \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) + \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right). \tag{21}$$

In matrix notation Eq. (20) can be written as

$$\begin{aligned}
\mathbf{M}^* &= \mathbf{M} + \begin{bmatrix} \boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) \\ \boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right) \\ \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)' \end{bmatrix}' \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{\eta}{1+\eta k_i} & 0 \\ 0 & 0 & 0 & \dfrac{-\eta}{1+\eta k_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) \\ \boldsymbol{f}'\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right) \\ \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)' \end{bmatrix} \\
&= \mathbf{M} + \mathbf{U}_1'\mathbf{D}_1\mathbf{U}_1,
\end{aligned} \tag{22}$$

where

$$\mathbf{U}_1 = \begin{bmatrix} \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) & \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}^*\right) & \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right) & \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right) \end{bmatrix}'$$

and

$$\mathbf{D}_1 = \mathrm{diag}\left(-1, 1, \frac{\eta}{1+\eta k_i}, -\frac{\eta}{1+\eta k_i}\right).$$

The updated information matrix is now of the form specified in Eq. (17), with $\mathbf{M}$ and $\mathbf{D}_1$ nonsingular $p \times p$ and 4×4 matrices, respectively, and $\mathbf{U}_1'$ and $\mathbf{U}_1 p \times 4$ and $4 \times p$ matrices, respectively. So, the expressions (18) and (19) can be used for computing $|\mathbf{M}^*|$ and $\mathbf{M}^{*-1}$, which results in the following update formulas:

$$\begin{aligned}
\left|\mathbf{M}^*\right| &= |\mathbf{M}|\,\left|\mathbf{I}_4 + \mathbf{D}_1\mathbf{U}_1\mathbf{M}^{-1}\mathbf{U}_1'\right| \\
&= |\mathbf{M}|\,|\mathbf{D}_1|\,\left|\mathbf{D}_1^{-1} + \mathbf{U}_1\mathbf{M}^{-1}\mathbf{U}_1'\right|
\end{aligned} \tag{23}$$

and

$$\mathbf{M}^{*-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}_1'\left(\mathbf{D}_1^{-1} + \mathbf{U}_1\mathbf{M}^{-1}\mathbf{U}_1'\right)^{-1}\mathbf{U}_1\mathbf{M}^{-1}, \tag{24}$$

where

$$|\mathbf{D}_1| = \frac{\eta^2}{(1+\eta k_i)^2}. \tag{25}$$

When looking at Eqs. (23) and (24) it should be clear that to compute the determinant or the inverse of the new information matrix, the calculation of the determinant or the inverse of a 4×4 matrix is required instead of a $p \times p$ matrix.

4. Changes to hard-to-change factor levels

As with those from the previous section the update formulas presented here are useful for point-exchange as well as coordinate-exchange algorithms. The major difference is that a change of the level of one or more hard-to-change factors in the jth run of the ith whole plot implies that every run in the ith whole plot must implement this change. The reason is that the level of every hard-to-change factor has to be held constant for all the runs in a whole plot. For this situation the updated information matrix becomes

$$\mathbf{M}^* = \mathbf{M} - \mathbf{X}_i'\mathbf{X}_i + \mathbf{X}_i^{*\prime}\mathbf{X}_i^* + \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' - \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)', \tag{26}$$

where

$$\mathbf{X}_i^* = \left[\boldsymbol{f}\left(\boldsymbol{w}_i^*, \boldsymbol{t}_{i1}\right)\ldots\boldsymbol{f}\left(\boldsymbol{w}_i^*, \boldsymbol{t}_{ik_i}\right)\right]', \tag{27}$$

with $\boldsymbol{w}_i^*$ the modified version of the hard-to-change factor levels $\boldsymbol{w}_i$ in whole plot i. In matrix notation, the information matrix in (26) can be written as

$$\begin{aligned}\mathbf{M}^* &= \mathbf{M} + \begin{bmatrix}\mathbf{X}_i \\ \mathbf{X}_i^* \\ \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_ik\right)'\end{bmatrix}' \begin{bmatrix}-\mathbf{I}_{k_i} & 0 & 0 & 0 \\ 0 & \mathbf{I}_{k_i} & 0 & 0 \\ 0 & 0 & \dfrac{\eta}{1+\eta k_i} & 0 \\ 0 & 0 & 0 & \dfrac{-\eta}{1+\eta k_i}\end{bmatrix} \begin{bmatrix}\mathbf{X}_i \\ \mathbf{X}_i^* \\ \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)' \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)'\end{bmatrix} \\ &= \mathbf{M} + \mathbf{U}_2'\mathbf{D}_2\mathbf{U}_2,\end{aligned} \tag{28}$$

where

$$\mathbf{U}_2 = \left[\mathbf{X}_i' \quad \mathbf{X}_i^{*\prime} \quad \left(\mathbf{X}_i'\mathbf{1}_{k_i}\right) \quad \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)\ \right]'$$

and

$$\mathbf{D}_2 = \operatorname{diag}\left(-\mathbf{I}_{k_i}, \mathbf{I}_{k_i}, \frac{\eta}{1+\eta k_i}, \frac{-\eta}{1+\eta k_i}\right).$$

All this results in the following update formulas:

$$\begin{aligned}\left|\mathbf{M}^*\right| &= |\mathbf{M}|\left|\mathbf{I}_{2k_i+2} + \mathbf{D}_2\mathbf{U}_2\mathbf{M}^{-1}\mathbf{U}_2'\right| \\ &= |\mathbf{M}|\,|\mathbf{D}_2|\left|\mathbf{D}_2^{-1} + \mathbf{U}_2\mathbf{M}^{-1}\mathbf{U}_2'\right|\end{aligned} \tag{29}$$

and

$$\mathbf{M}^{*-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}_2'\left(\mathbf{D}_2^{-1} + \mathbf{U}_2\mathbf{M}^{-1}\mathbf{U}_2'\right)^{-1}\mathbf{U}_2\mathbf{M}^{-1}, \tag{30}$$

where

$$|\mathbf{D}_2| = \frac{\eta^2}{(1+\eta k_i)^2}(-1)^{k_i+1}. \tag{31}$$

Here, $\mathbf{U}_2$ is a $(2k_i+2)\times p$ matrix and $\mathbf{D}_2$ is a diagonal $(2k_i+2)\times(2k_i+2)$ matrix. As a consequence, updating the determinant (or the inverse) of the information matrix after a change in one or more hard-to-change variables requires the computation of the determinant (or the inverse) of a $(2k_i+2)\times(2k_i+2)$ matrix which is advantageous when k_i is small in comparison to the number of model parameters p.

5. Swap of points between two whole plots

In the course of a point-exchange algorithm, such as the one by Goos and Vandebroek (2003), interchanges or swaps of two points from different whole plots i and l can be considered. For this to make sense, the levels of the hard-to-change factors should be the same for both points, i.e. $\boldsymbol{w}_i = \boldsymbol{w}_l$. When this is the case, a fast update formula for the determinant and the inverse can be derived too. The derivation uses the fact that the swap only has an impact on the corresponding rows of the submatrices $\mathbf{X}_i$ and $\mathbf{X}_l$ but not on the matrix product $\mathbf{X}'\mathbf{X}$. The updated information matrix equals

$$\begin{aligned}\mathbf{M}^* &= \mathbf{M} + \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i'\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i\mathbf{1}_{k_i}\right)' + \frac{\eta}{1+\eta k_l}\left(\mathbf{X}_l'\mathbf{1}_{k_l}\right)\left(\mathbf{X}_l'\mathbf{1}_{k_l}\right)' \\ &\quad - \frac{\eta}{1+\eta k_i}\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)' - \frac{\eta}{1+\eta k_l}\left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l}\right)\left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l}\right)',\end{aligned} \tag{32}$$

where $\mathbf{X}_i^*$ and $\mathbf{X}_l^*$ represent the updated versions of $\mathbf{X}_i$ and $\mathbf{X}_l$, respectively. When $\boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)$ and $\boldsymbol{f}\left(\boldsymbol{w}_l, \boldsymbol{t}_{lm}\right)$ denote the runs that are moved from the ith whole plot to the lth and vice versa, $\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}$ and $\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l}$ can easily be calculated as

$$\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i} = \mathbf{X}_i^{\prime}\mathbf{1}_{k_i} - \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) + \boldsymbol{f}\left(\boldsymbol{w}_l, \boldsymbol{t}_{lm}\right)$$

and

$$\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l} = \mathbf{X}_l^{\prime}\mathbf{1}_{k_l} - \boldsymbol{f}\left(\boldsymbol{w}_l, \boldsymbol{t}_{lm}\right) + \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right).$$

Eq. (32) can also be expressed as

$$\begin{aligned}
\mathbf{M}^* &= \mathbf{M} + \begin{bmatrix} \left(\mathbf{X}_i^{\prime}\mathbf{1}_{k_i}\right)^{\prime} \\ \left(\mathbf{X}_l^{\prime}\mathbf{1}_{k_l}\right)^{\prime} \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)^{\prime} \\ \left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l}\right)^{\prime} \end{bmatrix}^{\prime} \begin{bmatrix} \frac{\eta}{1+\eta k_i} & 0 & 0 & 0 \\ 0 & \frac{\eta}{1+\eta k_l} & 0 & 0 \\ 0 & 0 & \frac{-\eta}{1+\eta k_i} & 0 \\ 0 & 0 & 0 & \frac{-\eta}{1+\eta k_l} \end{bmatrix} \begin{bmatrix} \left(\mathbf{X}_i^{\prime}\mathbf{1}_{k_i}\right)^{\prime} \\ \left(\mathbf{X}_l^{\prime}\mathbf{1}_{k_l}\right)^{\prime} \\ \left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}\right)^{\prime} \\ \left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l}\right)^{\prime} \end{bmatrix} \\
&= \mathbf{M} + \mathbf{U}_3^{\prime}\mathbf{D}_3\mathbf{U}_3, \qquad (33)
\end{aligned}$$

where

$$\mathbf{U}_3 = \left[(\mathbf{X}_i^{\prime}\mathbf{1}_{k_i}) \quad (\mathbf{X}_l^{\prime}\mathbf{1}_{k_l}) \quad (\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i}) \quad (\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l})\right]^{\prime}$$

and

$$\mathbf{D}_3 = \operatorname{diag}\left(\frac{\eta}{1+\eta k_i}, \frac{\eta}{1+\eta k_l}, \frac{-\eta}{1+\eta k_i}, \frac{-\eta}{1+\eta k_l}\right).$$

This results in the following update formulas for the new information matrix:

$$\begin{aligned}
\left|\mathbf{M}^*\right| &= |\mathbf{M}| \left|\mathbf{I}_4 + \mathbf{D}_3\mathbf{U}_3\mathbf{M}^{-1}\mathbf{U}_3^{\prime}\right| \\
&= |\mathbf{M}|\,|\mathbf{D}_3| \left|\mathbf{D}_3^{-1} + \mathbf{U}_3\mathbf{M}^{-1}\mathbf{U}_3^{\prime}\right| \qquad (34)
\end{aligned}$$

and

$$\mathbf{M}^{*-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}_3^{\prime}\left(\mathbf{D}_3^{-1} + \mathbf{U}_3\mathbf{M}^{-1}\mathbf{U}_3^{\prime}\right)^{-1}\mathbf{U}_3\mathbf{M}^{-1}, \qquad (35)$$

where

$$|\mathbf{D}_3| = \left(\frac{\eta}{1+\eta k_i}\right)^2 \left(\frac{\eta}{1+\eta k_l}\right)^2.$$

As in Section 3, using the update formulas requires the calculation of the determinant or the inverse of a 4×4 matrix instead of a $p \times p$ matrix.

6. Change in the number of runs per whole plot

Update formulas can be derived for problems where the number of runs per whole plot is fixed, but also for problems where the number of runs in a whole plot can change in the course of the algorithm. A scenario in which the number of runs in a whole plot changes is when a run $\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)$ is removed from whole plot i and a new run, $\left(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}\right)$, is added to whole plot l. For this scenario the new information matrix can be written as

$$\begin{aligned}
\mathbf{M}^* &= \mathbf{M} - \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)\boldsymbol{f}^{\prime}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right) + \boldsymbol{f}\left(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}\right)\boldsymbol{f}^{\prime}\left(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}\right) \\
&\quad + \frac{\eta}{1+\eta k_i}\left[\left(\mathbf{X}_i^{\prime}\mathbf{1}_{k_i}\right)\left(\mathbf{X}_i^{\prime}\mathbf{1}_{k_i}\right)^{\prime}\right] - \frac{\eta}{1+\eta(k_i-1)}\left[\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1}\right)\left(\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1}\right)^{\prime}\right] \\
&\quad + \frac{\eta}{1+\eta k_l}\left[\left(\mathbf{X}_l^{\prime}\mathbf{1}_{k_l}\right)\left(\mathbf{X}_l^{\prime}\mathbf{1}_{k_l}\right)^{\prime}\right] - \frac{\eta}{1+\eta(k_l+1)}\left[\left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1}\right)\left(\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1}\right)^{\prime}\right], \qquad (36)
\end{aligned}$$

with

$$\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1} = \mathbf{X}_i^{\prime}\mathbf{1}_{k_i} - \boldsymbol{f}\left(\boldsymbol{w}_i, \boldsymbol{t}_{ij}\right)$$

and

$$\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1} = \mathbf{X}_l^{\prime}\mathbf{1}_{k_l} + \boldsymbol{f}\left(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}\right).$$

In matrix notation, Eq. (36) becomes

$$\mathbf{M}^* = \mathbf{M} + \begin{bmatrix} \boldsymbol{f}'(\boldsymbol{w}_i, \boldsymbol{t}_{ij}) \\ \boldsymbol{f}'(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}) \\ (\mathbf{X}_i'\mathbf{1}_{k_i})' \\ (\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1})' \\ (\mathbf{X}_l'\mathbf{1}_{k_l})' \\ (\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1})' \end{bmatrix}' \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\eta}{1+\eta k_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-\eta}{1+\eta(k_i-1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\eta}{1+\eta k_l} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-\eta}{1+\eta(k_l+1)} \end{bmatrix}$$

$$\times \begin{bmatrix} \boldsymbol{f}'(\boldsymbol{w}_i, \boldsymbol{t}_{ij}) \\ \boldsymbol{f}'(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1}) \\ (\mathbf{X}_i'\mathbf{1}_{k_i})' \\ (\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1})' \\ (\mathbf{X}_l'\mathbf{1}_{k_l})' \\ (\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1})' \end{bmatrix}. \tag{37}$$

This can also be written as

$$\mathbf{M}^* = \mathbf{M} + \mathbf{U}_4'\mathbf{D}_4\mathbf{U}_4, \tag{38}$$

with

$$\mathbf{U}_4 = \left[\boldsymbol{f}(\boldsymbol{w}_i, \boldsymbol{t}_{ij})\, \boldsymbol{f}(\boldsymbol{w}_l, \boldsymbol{t}_{l,k_l+1})\, (\mathbf{X}_i'\mathbf{1}_{k_i})\, (\mathbf{X}_i^{*\prime}\mathbf{1}_{k_i-1})\, (\mathbf{X}_l'\mathbf{1}_{k_l})\, (\mathbf{X}_l^{*\prime}\mathbf{1}_{k_l+1})\right]'$$

and

$$\mathbf{D}_4 = \mathrm{diag}\left(-1, 1, \frac{\eta}{1+\eta k_i}, \frac{-\eta}{1+\eta(k_i-1)}, \frac{\eta}{1+\eta k_l}, \frac{-\eta}{1+\eta(k_l+1)}\right).$$

This results in the following update formulas:

$$\begin{aligned} |\mathbf{M}^*| &= |\mathbf{M}|\,|\mathbf{I}_6 + \mathbf{D}_4\mathbf{U}_4\mathbf{M}^{-1}\mathbf{U}_4'| \\ &= |\mathbf{M}|\,|\mathbf{D}_4|\,|\mathbf{D}_4^{-1} + \mathbf{U}_4\mathbf{M}^{-1}\mathbf{U}_4'| \end{aligned} \tag{39}$$

and

$$\mathbf{M}^{*-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}_4'\left(\mathbf{D}_4^{-1} + \mathbf{U}_4\mathbf{M}^{-1}\mathbf{U}_4'\right)^{-1}\mathbf{U}_4\mathbf{M}^{-1}, \tag{40}$$

where

$$|\mathbf{D}_4| = \frac{-\eta^4}{(1+\eta k_i)(1+\eta k_l)\{1+\eta(k_i-1)\}\{1+\eta(k_l+1)\}}. \tag{41}$$

It is clear from Eqs. (39) and (40) that using the update formulas for the new information matrix requires the computation of the determinant or the inverse of a 6×6 matrix instead of a $p \times p$ matrix.

7. Discussion

To quantify the benefit of the presented update formulas for the algorithmic construction of optimal split-plot designs, a small simulation study was carried out. In this study a point-exchange algorithm was used to construct D-optimal two-level split-plot designs with different numbers of factors and whole plots to estimate a main-effects model. A point-exchange algorithm without any update formula was compared to a point-exchange algorithm using the update formulas (23) and (24) for changes to easy-to-change factor levels as well as the update formulas (29) and (30) for changes to hard-to-change factor levels. Next, the efficiency of the update formulas (34) and (35) for a swap of points between two whole plots was examined. Therefore, two point-swap algorithms were compared, one using the update formulas and one not using them.

While the use of the update formulas does not speed up the computations when the number of factors is as small as three, it does lead to substantial savings in computing time if the number of factors is four or more. Using the update formulas for the hard-to-change and easy-to-change factors speeds up the computations by a factor of four to six for a problem involving four factors, and by a factor of 32 to 90 for a six-factor split-plot design. The speed of the swapping algorithm is increased with a factor of five to eight for a problem involving four factors, and with a factor of around 200 for a six-factor problem.

As mentioned in the introduction, the model typically used in the context of blocked experiments with random block effects is the same as the one for split-plot experiments. As a result, the update formulas derived here are also useful for

Table 1
6-run suboptimal split-plot design.

Whole plot	w	t_1	t_2
1	−1	1	0
	−1	0	−1
2	0	1	0
	0	1	−1
3	−1	−1	−1
	−1	0	−1

Table 2
New intermediate 6-run split-plot design in scenario 1.

Whole plot	w	t_1	t_2
1	−1	1	0
	−1	0	−1
2	0	**−1**	**1**
	0	1	−1
3	−1	−1	−1
	−1	0	−1

the construction of optimal designs for blocked experiments. Only the results from Section 4 do not translate to blocked experiments because this kind of experiment does not involve hard-to-change factors. As a matter of fact, it is implicitly assumed that all factors in a blocked experiment are easy-to-change. Basically, to rewrite the update formulas for blocked experiments, all that has to be done is replace $\boldsymbol{f}(\boldsymbol{w}_i, \boldsymbol{t}_{ij})$ by $\boldsymbol{f}(\boldsymbol{t}_{ij})$.

This article is the first to discuss fast update formulas for changes to hard-to-change factor levels and for changes that affect the size of the whole plots (or blocks) in a split-plot (or a block) design. For changes to easy-to-change factor levels and swaps of points from different whole plots (or blocks), Goos and Vandebroek (2001a) mention that they used the update formula

$$\left|\mathbf{M} + \boldsymbol{u}\boldsymbol{u}'\right| = |\mathbf{M}|\,(1 + \boldsymbol{u}'\mathbf{M}^{-1}\boldsymbol{u}),$$

where $\boldsymbol{u}$ is a p-dimensional vector and $\mathbf{M}$ is the information matrix, in their point-exchange algorithm, without discussing the actual implementation. The problem with using just this update formula is that the calculation of $|\mathbf{M}|$ after an exchange of a design point with a candidate point and after the swap of two design points from different whole plots (or blocks) always requires updating the inverse of M. The new formulas provided in our paper render this computationally intensive extra update of $\mathbf{M}^{-1}$ unnecessary.

Appendix. A simple numerical example

In this Appendix, we present a simple numerical example to illustrate the application of the update formulas derived in Sections 3–6. We consider a small design with three factors of which w is the hard-to-change factor and t_1 and t_2 are the easy-to-change factors. We assume that a point-exchange algorithm is used to find a D-optimal split-plot design with six runs and three whole plots of two runs to estimate a main-effects model. Hence, $n = 6$, $k_1 = k_2 = k_3 = 2$ and $p = 4$. When using the point of a 3^3 factorial design as the set of candidate design points, a possible intermediate suboptimal design produced in the course of the point-exchange algorithm is given in Table 1. We now show what kind of exchanges and swaps can be made next to attempt to improve the design.

A.1. Scenario 1: Changes to easy-to-change factor levels

A possible modification of the design in Table 1 in the course of the algorithm could be the replacement of the third run, i.e. (0, 1, 0), by another point of the candidate set, say point (0, −1, 1). The new intermediate design, shown in Table 2, is then obtained. Since there is only a change in the levels of the easy-to-change factors t_1 and t_2, the update formulas in Section 3 are applicable. In this specific situation the change affects the first run of the second whole plot. Therefore, in Eqs. (20)–(25), $i = 2$ and $j = 2$,

$$\boldsymbol{f}(w_2, \boldsymbol{t}_{21}) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{f}(w_2, \boldsymbol{t}_{21}^{*}) = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 1 \end{bmatrix},$$

$$\mathbf{X}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}$$

Table 3
New intermediate 6-run split-plot design in scenario 2.

Whole plot	w	t_1	t_2
1	−1	1	0
	−1	0	−1
2	**1**	1	0
	1	1	−1
3	−1	−1	−1
	−1	0	−1

and

$$\mathbf{X}_2^* = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 0 & 1 & -1 \end{bmatrix}.$$

As a result,

$$\mathbf{U}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 1 \\ 2 & 0 & 2 & -1 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{D}_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{\eta}{1+2\eta} & 0 \\ 0 & 0 & 0 & \dfrac{-\eta}{1+2\eta} \end{bmatrix}.$$

A.2. Scenario 2: Changes to hard-to-change factor levels

Suppose that when reaching the suboptimal design in Table 1, the algorithm's next move is the replacement of the third run, i.e. (0, 1, 0), by point (1, 1, 0) from the candidate set. This move involves a change of the level of the hard-to-change factor w. Since the fourth run, i.e. (0, 1, −1), is executed in the same whole plot, its whole-plot factor level also has to change to 1, so that the fourth run equals (1, 1, −1) after the modification. The resulting new design is shown in Table 3. As all the changes take place in the second whole plot, $i = 2$. Furthermore

$$\mathbf{X}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}, \qquad \mathbf{X}_2^* = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix},$$

$$\mathbf{U}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & -1 \\ 2 & 0 & 2 & -1 \\ 2 & 2 & 2 & -1 \end{bmatrix}$$

and

$$\mathbf{D}_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\eta}{1+2\eta} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{-\eta}{1+2\eta} \end{bmatrix}.$$

A.3. Scenario 3: Swap of points between two whole plots

Since the level of the hard-to-change factor w in the suboptimal design of Table 1 is the same for the runs in whole plot one and three, another possible move could be the swap of the second run of the first whole plot ($i = 1, j = 2$) and the first run of the third whole plot ($l = 3, m = 1$), i.e. points (−1, 0, −1) and (−1, −1, −1), respectively. This way, the design in

Table 4
New intermediate 6-run split-plot design in scenario 3.

Whole plot	w	t_1	t_2
1	−1	1	0
	−1	**−1**	**−1**
2	0	1	0
	0	1	−1
3	**−1**	**0**	**−1**
	−1	0	−1

Table 5
New intermediate 6-run split-plot design in scenario 4.

Whole plot	w	t_1	t_2
1	−1	0	−1
2	0	1	0
	0	1	−1
	0	**−1**	**1**
3	−1	−1	−1
	−1	0	−1

Table 4 is obtained. The submatrices $\mathbf{X}_1$ and $\mathbf{X}_3$ and their updated versions, necessary the calculate the update information matrix in Eq. (31), are

$$\mathbf{X}_1 = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{X}_1^* = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & -1 & -1 & -1 \end{bmatrix},$$

$$\mathbf{X}_3 = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & -1 & 0 & -1 \end{bmatrix}$$

and

$$\mathbf{X}_3^* = \begin{bmatrix} 1 & -1 & 0 & -1 \\ 1 & -1 & 0 & -1 \end{bmatrix}.$$

In this case,

$$\mathbf{U}_3 = \begin{bmatrix} 2 & -2 & 1 & -1 \\ 2 & -2 & -1 & -2 \\ 2 & -2 & 0 & -1 \\ 2 & -2 & 0 & -2 \end{bmatrix}$$

and

$$\mathbf{D}_3 = \begin{bmatrix} \frac{\eta}{1+2\eta} & 0 & 0 & 0 \\ 0 & \frac{\eta}{1+2\eta} & 0 & 0 \\ 0 & 0 & \frac{-\eta}{1+2\eta} & 0 \\ 0 & 0 & 0 & \frac{-\eta}{1+2\eta} \end{bmatrix}.$$

A.4. Scenario 4: Change in the number of runs per whole plot

Suppose that the first run of the first whole plot, i.e. $(-1, 1, 0)$, is removed from the design in Table 1 and that the new point $(0, -1, 1)$ is added as a third run to the second whole plot. This leads to the design in Table 5. In this situation $i = j = 1$ and $l = 2$,

$$\boldsymbol{f}(w_1, \boldsymbol{t}_{11}) = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{f}(w_2, \boldsymbol{t}_{23}) = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 1 \end{bmatrix},$$

$$\mathbf{X}_1 = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 \end{bmatrix},$$
$$\mathbf{X}_1^* = \begin{bmatrix} 1 & -1 & 0 & -1 \end{bmatrix},$$
$$\mathbf{X}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}$$

and

$$\mathbf{X}_2^* = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 \\ 1 & 0 & -1 & 1 \end{bmatrix}.$$

The corresponding matrices $\mathbf{U}_4$ and $\mathbf{D}_4$ to update the information matrix, as shown in Eq. (38), are

$$\mathbf{U}_4 = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 1 & 0 & -1 & 1 \\ 2 & -2 & 1 & -1 \\ 1 & -1 & 0 & -1 \\ 2 & 0 & 2 & -1 \\ 3 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{D}_4 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{\eta}{1+2\eta} & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{-\eta}{1+\eta} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\eta}{1+2\eta} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{-\eta}{1+3\eta} \end{bmatrix}.$$

References

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

Eccleston, J.A., 1980. Recursive techniques for the construction of experimental designs. Journal of Statistical Planning and Inference 4, 291–297.

Goos, P., Vandebroek, M., 2001a. D-optimal response surface designs in the presence of random block effects. Computational Statistics and Data Analysis 37, 433–453.

Goos, P., Vandebroek, M., 2001b. Optimal split-plot designs. Journal of Quality Technology 33, 436–450.

Goos, P., Vandebroek, M., 2003. D-optimal split-plot designs with given numbers and sizes of whole plots. Technometrics 45, 235–245.

Goos, P., Vandebroek, M., 2004. Outperforming completely randomized designs. Journal of Quality Technology 36, 12–26.

Harville, D.A., 1997. Matrix Algebra from a Statistician's Perspective. Springer-Verlag, Telos.

John, J.A., 2001. Updating formula in an analysis of variance model. Biometrika 88, 1175–1178.

John, J.A., Whitaker, D., 2000. Recursive formulae for the average efficiency factor in block and row-column designs. Journal of the Royal Statistical Society Series B 62, 575–583.

John, J.A., Williams, E.R., 2000. Updating the average efficiency factor in -designs. Biometrika 87, 695–699.

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

Jones, B., Goos, P., 2007. A candidate-set-free algorithm for generating D-optimal split-plot designs. Applied Statistics 56, 347–364.

Kessels, R., Goos, P., Vandebroek, M., 2008. Optimal designs for conjoint experiments. Computational Statistics and Data Analysis 52, 2369–2387.

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

Nguyen, N.K., 1994. Construction of optimal block designs by computer. Technometrics 36, 300–307.

Nguyen, N.K., Liu, M.Q., 2008. An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays. Computational Statistics and Data Analysis 52 (2), 5269–5276.

Nguyen, N.K., Miller, A.J., 1992. A review of some exchange algorithms for constructing discrete D-optimal designs. Computational Statistics and Data Analysis 14, 489–498.

Nguyen, N.K., Williams, E.R., 2006. Experimental designs for 2-colour cdna microarray experiments. Applied Stochastic Models in Business and Industry 22, 631–638.