Journal of Statistical Planning and Inference 139 (2009) 30 – 44

Contents lists available at ScienceDirect

Journal of Statistical Planning and Inference

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

A variable-neighbourhood search algorithm for finding optimal run orders in the presence of serial correlation

Jean-Jacques Garroi[a], Peter Goos[a,*], Kenneth Sörensen[b,c]

[a]*Departement Wiskunde, Statistiek en Actuariële Wetenschappen, Universiteit Antwerpen, Belgium*
[b]*Centrum voor Industrieel Beleid, Katholieke Universiteit Leuven, Belgium*
[c]*Departement Milieu, Technologie en Technologiemanagement, Universiteit Antwerpen, Belgium*

ARTICLE INFO

Available online 23 May 2008

Keywords:
AR(1)
Autocorrelation
Central composite design
D-optimality criterion
Local search
Time trend

ABSTRACT

The responses obtained from response surface designs that are run sequentially often exhibit serial correlation or time trends. The order in which the runs of the design are performed then has an impact on the precision of the parameter estimators. This article proposes the use of a variable-neighbourhood search algorithm to compute run orders that guarantee a precise estimation of the effects of the experimental factors. The importance of using good run orders is demonstrated by seeking D-optimal run orders for a central composite design in the presence of an AR(1) autocorrelation pattern.

1. Introduction

Response surface designs, such as factorial and fractional factorial designs, central composite designs and Box–Behnken designs, are often run sequentially. Therefore, it is likely that the data exhibit patterns of serial correlation. In that case, the order in which the runs of the experimental design are carried out has an impact on the precision of the parameter estimates. This article presents a variable-neighbourhood search algorithm for finding efficient run orders for given designs.

The problem of finding run orders of standard experimental designs in the presence of serial correlation has received attention from Constantine (1989), who sought efficient run orders of weighing and factorial designs, Cheng and Steinberg (1991), who propose the reverse foldover algorithm to construct highly efficient run orders for two-level factorial designs, Martin et al. (1998b), who presented some results on two-level factorial designs, and Martin et al. (1998a), who discussed the construction of efficient run orders for multi-level factorial designs. Zhou (2001) presented a robust criterion for finding run orders of factorial and fractional factorial designs that are robust against the presence of serial correlation. All these authors concentrated on main-effects and main-effects-plus-interactions models.

A problem similar to the one considered in the present article is the quest for optimal run orders of experimental designs in the presence of deterministic time trends. Articles in that area focused on the construction of *trend-free* or *trend-robust* run orders, which ensure that the parameter estimates are not impacted by a time trend in the data. For example, Cheng (1990) and Cheng and Steinberg (1991) discussed trend-free run-orders for two-level factorial designs. A different line of research was started by Tack and Vandebroek (2001, 2002, 2003, 2004). Rather than using a given experimental design, they adopted an optimal design approach to select the design and to find an optimal run order simultaneously. As researchers often prefer to work with well-known standard designs, we adopt the same philosophy as Cheng (1990), Cheng and Steinberg (1991), and Martin et al. (1998a, b) and seek run orders for given designs in this article.

Although the focus in this article is on the search for optimal run orders in the presence of serial correlation, the presented algorithm is applicable to the problem of finding good run orders of a given design in the presence of time trends too. As the

* Corresponding author. Tel.: +32 3 220 40 59; fax: +32 3 220 48 17.
E-mail addresses: jean-jacques.garroi@ua.ac.be (J.-J. Garroi), peter.goos@ua.ac.be (P. Goos), kenneth.sorensen@cib.kuleuven.be (K. Sörensen).

0378-3758/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jspi.2008.05.014

problem of ordering the runs of factorial designs for efficiently estimating first-order models has been studied extensively, we illustrate our algorithm using a popular type of experimental design for estimating second-order response surface models, namely central composite designs.

Constantine (1989), Martin et al. (1998b), Cheng (1990), and Cheng and Steinberg (1991) show that a key feature of run orders for two-level factorial designs that ensure a precise estimation of the factor effects in the presence of serial correlation and time trends is that they require a large number of changes of the levels of the experimental factors. In this paper, we use the newly developed variable-neighbourhood search algorithm and Hamming distances to investigate to what extent this result carries over to optimal run orders for central composite designs in the presence of serial correlation.

The article is organized as follows. Section 2 provides a description of the statistical model and the design criteria considered in this paper. Section 3 serves as a motivation for this article and illustrates how large the impact of the use of different run orders on the precision of the parameter estimates can be. A variable-neighbourhood search algorithm for seeking efficient run orders is presented in Section 4. Finally, a selection of computational results and a description of the features of the optimal run orders are given in Section 5. A discussion including suggestions for future research concludes the paper.

2. Model and design criteria

The algorithm presented in this article is applied to the problem of finding optimal run orders for central composite designs. The model that is usually estimated when this type of experimental design is utilized is the full second-order model

$$Y_t = \beta_0 + \sum_{i=1}^{k} \beta_i x_{it} + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_{it} x_{jt} + \sum_{i=1}^{k} \beta_{ii} x_{it}^2 + \varepsilon_t,$$

where k represents the number of experimental factors, Y_t represents the response observed at run t, x_{it} is the level of the ith factor at that run, $\beta_0, \beta_1, \ldots, \beta_{kk}$ represent the $p = k(k-1)/2 + 2k + 1$ unknown model parameters, and ε_t is the random error at the tth run of the central composite design. The random errors ε_t, $t = 1, 2, \ldots, n$ with n the number of runs in the design, are assumed to have a zero mean and to follow an AR(1) correlation pattern. If we denote $[1\ x_{1t}\ \ldots\ x_{kt}^2]'$ by $\mathbf{x}_t$ and $[\beta_0\ \beta_1\ \ldots\ \beta_{kk}]'$ by $\boldsymbol{\beta}$, the model is given by

$$Y_t = \mathbf{x}_t' \boldsymbol{\beta} + \varepsilon_t,$$

where

$$\varepsilon_t = \rho \varepsilon_{t-1} + v_t,$$

$0 < \rho < 1$, and $\mathrm{E}(v_t) = 0$ and $\mathrm{var}(v_t) = \sigma_v^2$ for all t. In matrix notation, the model can be written as

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

where $\boldsymbol{Y} = [Y_1\ Y_2\ \ldots\ Y_n]'$, the model matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_n]'$ and $\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2\ \ldots\ \varepsilon_n]'$. The covariance matrix of $\boldsymbol{\varepsilon}$ and $\boldsymbol{Y}$ is

$$\mathbf{V} = \frac{\sigma_v^2}{1-\rho^2} \begin{bmatrix} 1 & \rho & \rho^2 & \ldots & \rho^{n-1} \\ \rho & 1 & \rho & \ldots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \ldots & \rho^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \ldots & 1 \end{bmatrix}.$$

Under these assumptions, the best linear unbiased estimator is the generalized least squares (GLS) estimator

$$\hat{\boldsymbol{\beta}}_{\mathrm{GLS}} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\boldsymbol{Y}, \tag{1}$$

with covariance matrix $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}$. Because of computational convenience or ignorance about the presence and nature of the serial correlation, many experimenters however will use the ordinary least squares (OLS) estimator

$$\hat{\boldsymbol{\beta}}_{\mathrm{OLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{Y}, \tag{2}$$

which has covariance matrix $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$.

This led us to consider two different criteria for selecting run orders. First, we seek run orders that maximize $n|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}|^{1/p}$. These run orders minimize the volume of the confidence ellipsoid about the unknown model parameters contained in $\boldsymbol{\beta}$ when the GLS estimator (1) is used. Second, we seek run orders that maximize $n|\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}|^{1/p}$. These run orders minimize the volume of the confidence ellipsoid about the unknown model parameters when the OLS estimator is applied. The two criteria are D-optimality criteria (Atkinson and Donev, 1992; Goos, 2002) which is why $n|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}|^{1/p}$ and $n|\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}|^{1/p}$ are called

D-optimality criterion values and which offers the technical advantage that the coding of the experimental factors does not have an impact on the optimal run order. The optimal run orders depend on the magnitude of the correlation coefficient ρ, but not on the variance σ_v^2. Without loss of generality, we therefore assume that $\sigma_v^2 = 1$ in this paper.

The quality of a given run order, with model matrix $\mathbf{X}$, is expressed using a D-efficiency. For situations in which GLS estimation is used, the D-efficiency of a certain run order with model matrix $\mathbf{X}$ is defined as

$$D_{\mathrm{GLS}} = \left\{ \frac{|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}|}{|\mathbf{X}'_{\mathrm{GLS}}\mathbf{V}^{-1}\mathbf{X}_{\mathrm{GLS}}|} \right\}^{1/p},$$

where $\mathbf{X}_{\mathrm{GLS}}$ represents the model matrix corresponding to the optimal run order for GLS estimation. For situations in which OLS estimation is performed, the D-efficiency of a certain run order with model matrix $\mathbf{X}$ is defined as

$$D_{\mathrm{OLS}} = \left\{ \frac{|\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}|}{|\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}(\mathbf{X}'_{\mathrm{OLS}}\mathbf{V}\mathbf{X}_{\mathrm{OLS}})^{-1}\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}|} \right\}^{1/p},$$

where $\mathbf{X}_{\mathrm{OLS}}$ is the model matrix corresponding to the optimal run order for OLS estimation. The loss of D-efficiency incurred by not using the optimal run orders is expressed as $(1 - D_{\mathrm{GLS}}) \times 100\%$ and $(1 - D_{\mathrm{OLS}}) \times 100\%$ for GLS and OLS, respectively.

3. The impact of the run order

Randomization is a keyword in standard textbooks on experimental design. For response surface designs that are run sequentially, this means, among other things, that the runs should be carried out in random orders. When the observations are statistically independent, for example when $\mathbf{V} = \sigma_\varepsilon^2 \mathbf{I}_n$, each of the run orders leads to the same value, $n\sigma_\varepsilon^{-2}|\mathbf{X}\mathbf{X}|^{1/p}$, for the two D-optimality criteria introduced in the previous section. However, in the presence of serial correlation, some run orders lead to substantially smaller values for the D-optimality criterion than others. We illustrate this by means of central composite designs.

3.1. Central composite designs

In general, central composite designs are composed of three types of experimental runs:

(1) 2^k or, for larger values of k, 2^{k-f} factorial points;
(2) $2k$ axial points at a distance α from the centre; and
(3) several centre points.

We denote the number of factorial, axial and centre points in the central composite design by n_{F}, n_{A} and n_{C}, respectively. The exact value of α used for the axial points depends on the shape of the experimental region, the number of experimental factors k and technical properties. Common choices for α are 1 (when the design region is cuboidal) and $\sqrt[4]{n_{\mathrm{F}}}$ when the goal is to have a rotatable design for a spherical region. Also the number of centre points utilized depends on the number of experimental factors and technicalities such as the potential need for blocking the design (Myers and Montgomery, 1995). A 17-run central composite design for three factors x_1, x_2 and x_3 is displayed in Table 1. The first eight points in the table are the factorial portion of the design, whereas the points 9–14 are the axial points. The last three points are centre runs. A graphical representation of the central composite design is given in Fig. 1. In this article, we use $\alpha = \sqrt[4]{n_{\mathrm{F}}}$ for each of the axial runs.

Table 1
Central composite design for three experimental factors

Run	Label	x_1	x_2	x_3
1	1	−1	−1	−1
2	2	−1	−1	+1
3	3	−1	+1	−1
4	4	−1	+1	+1
5	5	+1	−1	−1
6	6	+1	−1	+1
7	7	+1	+1	−1
8	8	+1	+1	+1
9	9	−α	0	0
10	10	+α	0	0
11	11	0	−α	0
12	12	0	+α	0
13	13	0	0	−α
14	14	0	0	+α
15	15	0	0	0
16	15	0	0	0
17	15	0	0	0

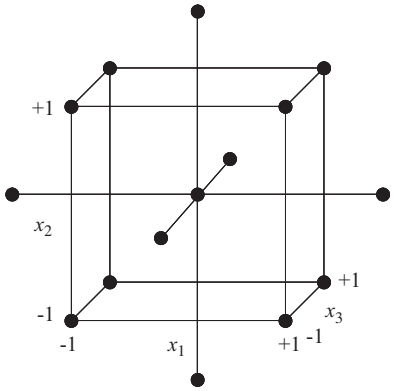

Fig. 1. Graphical representation of a three-factor central composite design with $\alpha = \sqrt[4]{n_F}$.

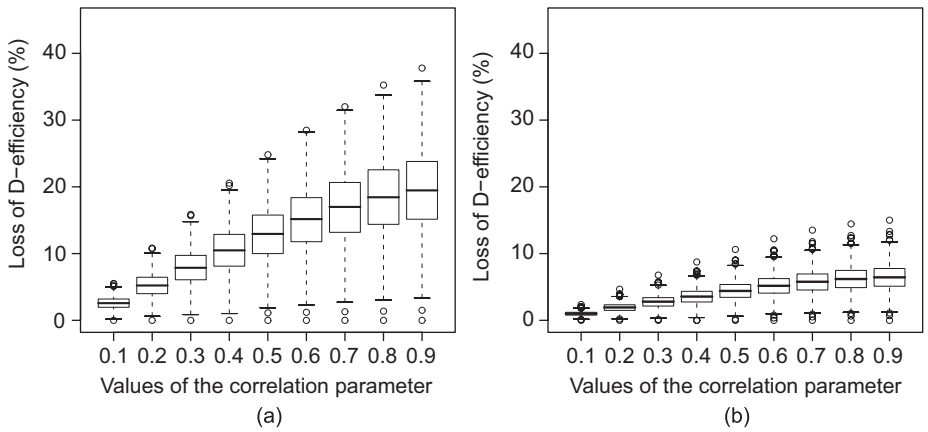

Fig. 2. Losses in D-efficiency for various values of the correlation parameter ρ when a random run order and generalized least squares estimation are used: (a) 3-factor central composite design; (b) 6-factor central composite design.

Many standard textbooks stress that central composite designs lend themselves to sequential experimentation: the factorial points are run first, followed by the centre points and then by the axial points. The factorial points are run first to check whether all the experimental factors are important. Next, the centre points are run to check for curvature in the response. Finally, if curvature is found, the axial points are run to be able to estimate the quadratic effects of the experimental factors. In this article, however, it is assumed that prior experimentation has learnt that all experimental factors are active and that curvature is present, so that the central composite design can be performed one experimental run at a time without restrictions on the order of the runs. The number of possible run orders of a central composite design with n_C centre runs then equals

$$n!/n_C! = (n_F + n_A + n_C)!/n_C! = (2^k + 2k + n_C)!/n_C!$$

when all factorial points are included in the design. This number increases very rapidly with k, the number of experimental factors. For k as small as two, and three centre runs, already 6,652,800 different run orders are possible. For six factors and eight centre points, more than 10^{121} run orders exist. These numbers make it impossible to enumerate all run orders in order to select the best one(s). In the next section, we investigate the range of losses in D-efficiency that can be obtained when random run orders are used for central composite designs with three and six experimental factors.

3.2. Random run orders

First, we generated 1000 random run orders of a central composite design involving three factors and three centre runs. Under the assumption that the correlation structure of the residuals is of the AR(1) type, we computed the losses in D-efficiency of each of these run orders with respect to the randomly generated order that produced the largest D-optimality criterion value for GLS estimation.

Side-by-side box-plots of the losses in D-efficiency for ρ values between 0.1 and 0.9 obtained in this way are displayed in Fig. 2a. The range of the losses clearly increases with ρ, indicating that the impact of the run order on the precision of the estimation

is especially important when the serial correlation between the responses is high. However, even for a ρ value as small as 0.1, losses in D-efficiency of 5% were observed. For larger correlations, the losses in D-efficiency rapidly become more substantial.

We also generated 1000 random run orders for a six-factor central composite design with two centre points. Side-by-side box-plots of the losses in D-efficiency for these run orders relative to the best randomly generated order are shown in Fig. 2b. It turns out that the losses in D-efficiency are substantially smaller for the six-factor design than for the three-factor design. For example, the maximum observed loss in D-efficiency was around 2% for a ρ value of 0.1. As can be observed in Fig. 2b, the magnitudes of the losses in D-efficiency again increase with ρ.

These computational results suggest that using random run orders is generally not efficient and that using optimal instead of random run orders would be appropriate. In the next section, we present an algorithmic approach to seeking optimal run orders of given experimental designs.

4. An algorithm for finding optimal run orders

Although there is a rich literature on the algorithmic construction of optimal experimental designs (see, for example, Goos, 2002), the problem of finding optimal run orders of standard second-order designs has not received attention in the literature. In this section, we outline the algorithm we propose for that purpose. The algorithm builds on local search algorithms developed for various applications in operational research and operations management.

4.1. Local search algorithms and encoding of solutions

In local-search-based algorithms for combinatorial optimization, the search for a good solution is done by iteratively making one small change (called a *move*) to the *current solution s*. All solutions s' that can be reached in one single move starting from a given solution s are said to be in the *neighbourhood* of that solution, denoted by $N(s)$. A *local optimum* is then defined as a solution that does not have any better solution in its neighbourhood, i.e. no improving moves can be made from this solution. Many metaheuristics have been developed, each using a different strategy to escape from these local optima and hopefully reach the *global optimum*. We define the size of a neighbourhood as the number of solutions it contains, i.e. the number of solutions that can be reached in a single move from the current solution. The size of a neighbourhood usually grows with the size of the problem n. The size n of the problem discussed in this paper is the number of runs or points included in the design.

One of the first choices that needs to be made when developing a metaheuristic is the *encoding* of the solution. Since our problem is to determine the best possible *order* in which to run a standard experimental design, an obvious choice is to use a permutation encoding. To this end, we assign a label to each point in the design and represent a solution as a permutation of these labels. This permutation may contain a repetition of labels, if some design points occur more than once. In the design problem discussed in this paper, the centre design point is repeated n_C times, so that the number of unique labels in each permutation equals $n_F + n_A + 1 = n - n_C + 1$. For example, the 15 labels we utilized for encoding the 17-point three-factor central composite design in Table 1 are shown in the table's second column. In general, the labels $1, 2, \ldots, n_F$ are assigned to the factorial points, the labels $n_F + 1, n_F + 2, \ldots, n_F + n_A$ are used for the axial points, and the label $n - n_C + 1$ is utilized for all centre points in the design.

Obviously, many possible "small changes" can be made to any solution. In this context, we should be careful to distinguish between a *move type* and a move itself. For example, if a solution is represented as a permutation of a set of items, a possible move type is "switch the items in two different positions". A possible move is "switch the items in positions 1 and 5". A move type gives rise to a certain *neighbourhood structure*. Given a move type and a solution, a complete neighbourhood can be derived.

4.2. Variable-neighbourhood search

Variable-neighbourhood search (Mladenović and Hansen, 1997) is a recent metaheuristic for combinatorial optimization. As its name suggests, this metaheuristic systematically explores different *neighbourhood structures*. The main idea underlying variable-neighbourhood search is that a local optimum relative to a certain neighbourhood structure not necessarily is a local optimum relative to another neighbourhood structure. For this reason, escaping from a local optimum can be done by changing the neighbourhood structure. Despite it being a relatively recent development, variable-neighbourhood search has been successfully applied to a wide variety of well-known and lesser-known optimization problems such as vehicle routing (Kytöjoki et al., 2007), project scheduling (Fleszar and Hindi, 2004), automatic discovery of theorems (Caporossi and Hansen, 2004), graph coloring (Avanthay et al., 2003) and the synthesis of radar polyphase codes (Mladenović et al., 2003).

Usually, variable-neighbourhood search algorithms examine neighbourhoods in a certain order, from small to large. The reason is that small neighbourhoods contain fewer solutions and can hence be searched in less time than large neighbourhoods. Larger neighbourhoods are only examined when all smaller neighbourhoods have been depleted, i.e. when the current solution is a local optimum of all smaller neighbourhoods. When a solution is found that is a local optimum of all neighbourhoods, a variable-neighbourhood search algorithm uses a *perturbation move* (called *shake* in the variable-neighbourhood literature) to jump to a different part of the solution space. The exploration of the different neighbourhoods is then continued from there.

Table 2
Neighbourhood structures used in the variable-neighbourhood search algorithm

N_k	Name	Size	Description	Example
N_1	Shift right	$O(n)$	Move all items 1 position to the right, move the last item to position 1, repeat $n-1$ times	[3,4,5,5,1,2]
N_2	Swap adjacent	$O(n)$	Swap any two items in adjacent positions	[1,2,4,3,5,5]
N_3	Relabel	$O(n)$	Use unique labels for all centre points, add 1 to each label, change label $n+1$ to 1 and labels $n_F+n_A+2,\ldots,n$ to n_F+n_A+1, repeat $n-1$ times	[2,3,4,5,5,1]
N_4	Swap	$O(n^2)$	Swap any two items	[1,5,3,4,5,2]
N_5	Move	$O(n^2)$	Move any item to any different position	[2,3,4,5,1,5]
N_6	2-Opt	$O(n^2)$	Reverse the solution between any two items	[1,5,4,3,2,5]

4.3. Neighbourhood structures and perturbation move

In our variable-neighbourhood search algorithm,[1] we use six different neighbourhoods, half of which have a size that increases proportionally with the size of the problem (i.e. linear neighbourhoods of size $O(n)$). The other half of the neighbourhoods have sizes that increase according to a second-order polynomial function of the design size (these are quadratic neighbourhoods or neighbourhoods of size $O(n^2)$). The neighbourhood structures used are summarized in Table 2. The table also contains an instance of each neighbourhood structure assuming the current solution s is given by [1,2,3,4,5,5], where label 5 corresponds to a centre point. For this example, n thus equals 6 and n_C is equal to 2.

To ensure that the number of centre points does not change, the *relabel* move type—contrary to the other move types—requires us to temporarily assign unique labels to all centre points. This is done by scanning the permutation from left to right and assigning labels $n-n_C+1$ to n to the n_C centre points encountered. For example, starting from solution [1,2,3,4,5,5], we first transform the original encoding into [1,2,3,4,5,6], add 1 to each label and change label 7 to 1, yielding [2,3,4,5,6,1]. The solution is then re-encoded to [2,3,4,5,5,1] by assigning label $n-n_C+1$ to all points having a label that exceeds this value.

The order of the neighbourhoods in our algorithm is such that the larger neighbourhoods are only used when all smaller ones have been exhausted. The order of neighbourhoods of equal size was chosen arbitrarily. In each neighbourhood, we use a *steepest ascent* move strategy, i.e. we generate all solutions in the neighbourhood, examine their quality and select the best one. This strategy ensures a fast increase of the objective function value.

The perturbation scheme we use is very simple: we perform a swap of two randomly selected items twice. This perturbation disturbs the solution enough to prevent the search from immediately converging to the previous local optimum, while maintaining most of the quality of the solution obtained by the previous iteration of the variable-neighbourhood search.

A pseudo-code outline of the algorithm is provided in Algorithm 1. In the outline, κ_{max} represents the number of neighbourhoods utilized and κ indicates which neighbourhood structure is currently searched by the algorithm.

Algorithm 1. VNS algorithm
Generate a random initial solution s;
repeat
  Set $\kappa \leftarrow 1$;
  while $\kappa \leqslant \kappa_{max}$ **do**
    Find the best solution $s' \in N_\kappa(s)$;
    if s' *better than* s **then**
      Set $s \leftarrow s'$;
      Set $\kappa \leftarrow 1$
    else
      Set $\kappa \leftarrow \kappa + 1$
  Perturb s;
until *stopping criterion*;

5. Computational results

Using the variable-neighbourhood search algorithm, we have searched optimal arrangements of several central composite designs. The algorithm was coded in R and run on a Pentium 4 2.8 GHz with 1 GB internal memory.

To compute the run orders displayed in this section, the algorithm was run for 50 iterations of the main loop (i.e. until 50 perturbations had been performed). As the best solution was usually found after only a few iterations, the value of 50 was judged

[1] It would be more precise to say that our algorithm is a variable neighbourhood *descent* (VND) method. VND is a deterministic variant of VNS that does not use a random "shaking" move before starting the local search.

Table 3
D-optimality criterion values of the D-optimal run orders for ordinary least squares (OLS) and generalized least squares (GLS) estimation

ρ	OLS	GLS
0.1	200.257262	201.269715
0.2	204.612429	208.641952
0.3	208.257348	217.304693
0.4	210.878225	226.979588
0.5	212.501700	237.379511
0.6	212.256481	247.600109
0.7	208.973890	256.385308
0.8	201.064133	261.573121
0.9	184.908149	257.121911

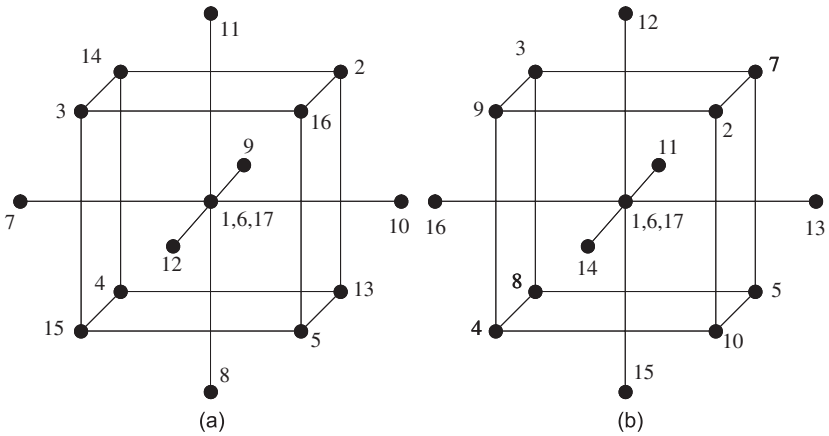

Fig. 3. D-optimal run orders for the GLS estimator when $0 < \rho \leqslant 0.39$.

to be more than enough. Detailed analysis of the operation of the algorithm shows that all neighbourhoods are indeed used when searching for the optimal solution, even though the linear neighbourhoods are used far more often. Run times of the algorithm were between 5 and 10 min for the 17-run central composite design in Table 1. Regarding these run times, it should be pointed out that no attempts were made to optimize the algorithm for speed, for example by using fast update procedures for the information matrix.

The results we display in this section are for the 17-run central composite design in Table 1. The D-optimality criterion values, $n|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}|^{1/p}$ for GLS and $n|\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}|^{1/p}$ for OLS, of the D-optimal run orders for this design are displayed in Table 3. The beneficial effect of taking into account the correlation pattern when estimating the model is visible through the fact that the D-optimality criterion values are larger for GLS estimation than for OLS estimation.

Even for a design as small as the 17-run central composite design, representing the computational results is difficult as the optimal run orders depend on the correlation parameter ρ and on the estimation method used. Moreover, more than one run order turned out to be D-optimal for any given combination of the correlation parameter and the estimation method. Only a portion of the optimal run orders is displayed here. It is in general difficult to explain how one optimal run order can be transformed into another optimal run order. One simple transformation, however, is to reverse the run order of a given optimal design. Another simple possibility is to switch the signs of the values for one or more experimental factors in a given optimal run order. The graphical representations of alternative run orders obtained by switching the signs of the levels of one or more factors would be nothing but mirror images or rotations of the originals.

5.1. Optimal run orders for GLS estimation

For GLS estimation, three sets of D-optimal run orders were found. The first set contains run orders that are D-optimal for ρ values between 0 and 0.39. Two run orders from that set are displayed in Fig. 3. The second set contains run orders that are D-optimal for ρ values between 0.39 and 0.72. Two run orders from that set are displayed in Fig. 4. Finally, the third set contains run orders that are D-optimal for ρ values above 0.72. Two run orders from that set are displayed in Fig. 5. In each of these figures, the values of x_1, x_2 and x_3 are assigned to the axes as indicated in Fig. 1. An alternative representation of the D-optimal run orders for the GLS estimator is given in Table 4. This representation uses the labels given in the second column of Table 1.

Although the run orders displayed in the three figures and Table 4 are substantially different, the first as well as the last run in each of them is a centre-point run. Also, sequences of axial runs can be observed in each of the optimal run orders. In these sequences, which can most easily be seen in Table 4, the axial points (which have labels 9–14) are arranged so that a

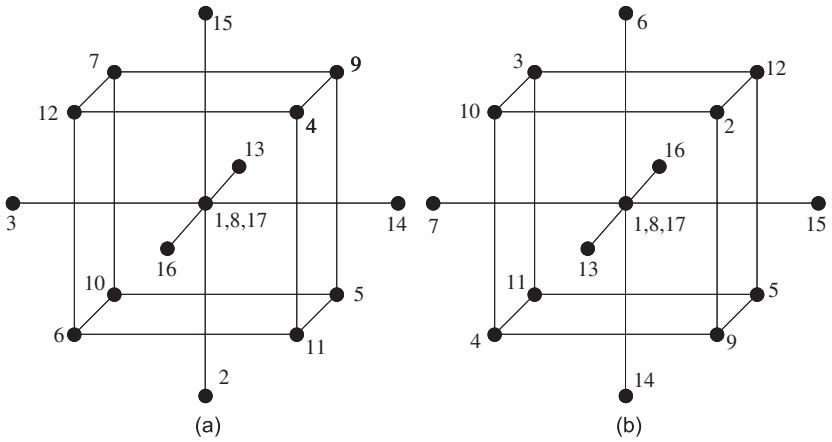

Fig. 4. D-optimal run orders for the GLS estimator when $0.39 \leqslant \rho \leqslant 0.72$.

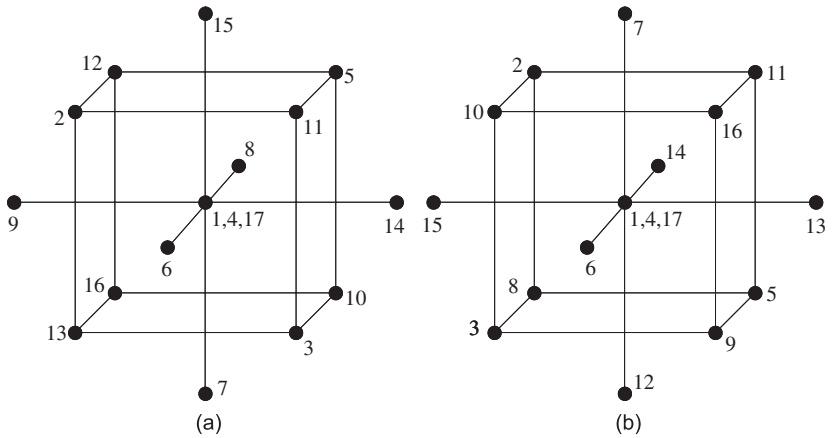

Fig. 5. D-optimal run orders for the GLS estimator when $0.72 \leqslant \rho$.

Table 4
D-optimal run orders for the GLS estimator

ρ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0.0, 0.39	15	8	3	2	5	15	9	11	14	10	12	13	6	4	1	7	15
	15	7	4	1	6	15	8	2	3	5	14	12	10	13	11	9	15
							8										
0.39–0.72	15	11	9	7	6	1	4	15	8	2	5	3	14	10	12	13	15
	15	7	4	1	6	12	9	15	5	3	2	8	13	11	10	14	15
							8										
0.72, 0.99	15	3	5	15	8	13	11	14	9	6	7	4	1	10	12	2	15
	15	4	1	15	6	13	12	2	5	3	8	11	10	14	9	7	15

The factorial, axial and centre points have labels 1–8, 9–14 and 15, respectively.

given axial point is never preceded or succeeded by its mirror image. Except for large values of ρ, also the factorial points of the central composite design appear in clusters. Within these clusters, the coordinates of which satisfy the equations $x_1x_2x_3 = -1$ or $x_1x_2x_3 = +1$, the points are arranged so that the values in the columns of the model matrix corresponding to the main effects and two-factor interactions are different in large numbers of pairs of consecutive runs. This result is consistent with the results of Constantine (1989), Cheng and Steinberg (1991), and Martin et al. (1998b) for two-level factorial designs and it is illustrated in Table 5 by the model matrix of the D-optimal run order displayed in Fig. 3a. The table also contains the Hamming distances, labelled HD, for each consecutive pair of rows of the model matrix. The Hamming (1950) distance between two vectors is widely used in information theory and is defined as the number of elements that differ between the two vectors. Using it here for each pair of consecutive runs allows us to quantify the extent to which the consecutive rows of the design matrices of the optimal run orders are different, and thus to check whether it is important to have a maximum number of changes or not.

Table 5
Model matrix for the D-optimal run order displayed in Fig. 3a

Run	I	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	x_1^2	x_2^2	x_3^2	HD
1	1	0	0	0	0	0	0	0	0	0	
2	1	1	1	1	1	1	1	1	1	1	9
3	1	−1	1	−1	−1	1	−1	1	1	1	4
4	1	−1	−1	1	1	−1	−1	1	1	1	4
5	1	1	−1	−1	−1	−1	1	1	1	1	4
6	1	0	0	0	0	0	0	0	0	0	9
7	1	$-\alpha$	0	0	0	0	0	α^2	0	0	2
8	1	0	$-\alpha$	0	0	0	0	0	α^2	0	4
9	1	0	0	α	0	0	0	0	0	α^2	4
10	1	α	0	0	0	0	0	α^2	0	0	4
11	1	0	α	0	0	0	0	0	α^2	0	4
12	1	0	0	$-\alpha$	0	0	0	0	0	α^2	4
13	1	1	−1	1	−1	1	−1	1	1	1	9
14	1	−1	1	1	−1	−1	1	1	1	1	4
15	1	−1	−1	−1	1	1	1	1	1	1	4
16	1	1	1	−1	1	−1	−1	1	1	1	4
17	1	0	0	0	0	0	0	0	0	0	9

The last column of the table contains the Hamming distances between consecutive rows.

Table 6
Model matrix for a run order with maximum Hamming distance between each pair of consecutive rows

Run	I	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	x_1^2	x_2^2	x_3^2	HD
1	1	0	0	0	0	0	0	0	0	0	
2	1	−1	−1	1	1	−1	−1	1	1	1	9
3	1	α	0	0	0	0	0	α^2	0	0	9
4	1	1	−1	−1	−1	1	−1	1	1	1	9
5	1	0	α	0	0	0	0	0	α^2	0	9
6	1	1	1	1	1	1	1	1	1	1	9
7	1	0	0	α	0	0	0	0	0	α^2	9
8	1	−1	1	−1	−1	−1	1	1	1	1	9
9	1	0	0	0	0	0	0	0	0	0	9
10	1	−1	−1	−1	1	1	1	1	1	1	9
11	1	$-\alpha$	0	0	0	0	0	α^2	0	0	9
12	1	1	1	−1	1	−1	−1	1	1	1	9
13	1	0	$-\alpha$	0	0	0	0	0	α^2	0	9
14	1	1	−1	1	−1	−1	1	1	1	1	9
15	1	0	0	$-\alpha$	0	0	0	0	0	α^2	9
16	1	−1	1	1	−1	1	−1	1	1	1	9
17	1	0	0	0	0	0	0	0	0	0	9

Within each cluster of factorial points, the Hamming distance between each pair of rows of the model matrix in Table 5 is 4. This is the maximum Hamming distance that can be achieved between the rows of any two factorial points in a design involving three factors. Similarly, the Hamming distance between each pair of consecutive rows within the cluster of axial points is always equal to four, which is the maximum Hamming distance that is possible between any pair of axial points.

The arrangement of the axial points and the factorial points in separate clusters however makes that the sum of all Hamming distances between consecutive rows in the model matrix is not at all maximized. We illustrate this by means of the model matrix in Table 5.

The maximum Hamming distance between any two consecutive runs of a 17-run three-factor central composite design is equal to nine because the column corresponding to the intercept is a column of ones. As a result, the maximum Hamming distance possible is $9 \times (17 - 1) = 144$. It is clear that this can be achieved only when (i) a centre point is followed by a factorial point, (ii) a factorial point is followed by a centre point, (iii) an axial point is followed by a factorial point (if $\alpha \neq 1$, as it is assumed in this article), or (iv) a factorial point is followed by an axial point (if $\alpha \neq 1$). In the D-optimal run order displayed in Table 5, this happens only four times, namely for runs 2, 6, 13 and 17. For all but one other pair of two consecutive runs, the Hamming distance equals four. Each of these 11 cases corresponds to a sequence of two factorial points or two axial points in the D-optimal run order. Finally, the axial point that follows the centre point in run 6 entails a Hamming distance of two only. This gives a total sum of 82 for the Hamming distances between all consecutive pairs of rows in the D-optimal model matrix in Table 5.

In the light of the results of Constantine (1989), Cheng and Steinberg (1991), and Martin et al. (1998a, b), who found that optimal arrangements of factorial designs require large numbers of level changes, it is surprising that the sum of the Hamming distances in the optimal run order in Table 5 is only 82. It can be shown, however, that designs with a maximum sum of Hamming distances are substantially inferior to the D-optimal run orders. Consider, for example, the run order displayed in Table 6. The model matrix corresponding to this run order has 144 level changes. Nevertheless, its D_{GLS}-efficiency relative to the D-optimal

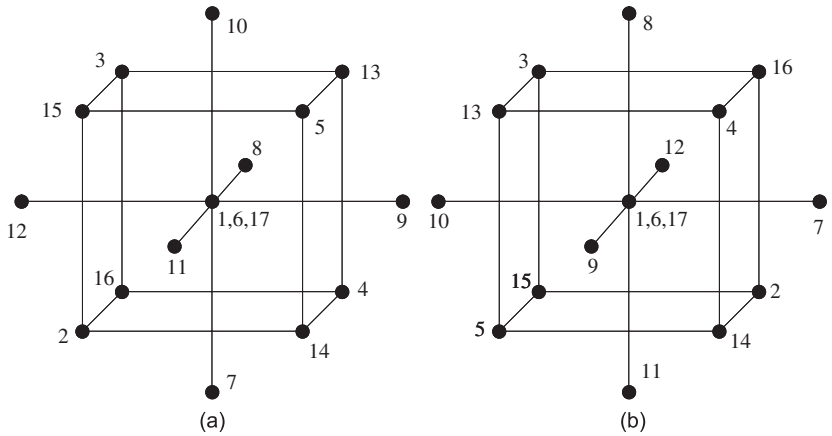

Fig. 6. D-optimal run orders for the OLS estimator when $0 < \rho \leqslant 0.11$.

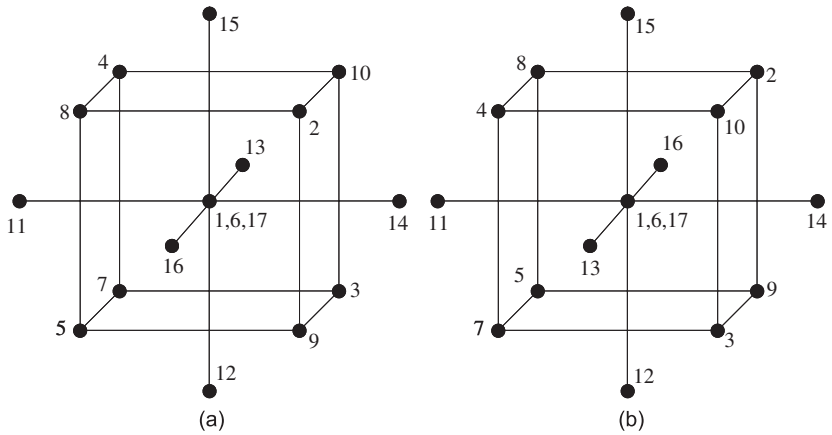

Fig. 7. D-optimal run orders for the OLS estimator when $0.11 \leqslant \rho \leqslant 0.41$.

run order in Table 5 is substantially smaller than one. For example, its D_{GLS}-efficiency is only 96.35% when ρ is 0.1 and 89.83% when ρ equals 0.3.

5.2. Optimal run orders for OLS estimation

For OLS estimation, four sets of D-optimal run orders were found. The first set contains run orders that are D-optimal for ρ values between 0 and 0.11. Two run orders from that set are displayed in Fig. 6. The second set contains run orders that are D-optimal for ρ values between 0.11 and 0.41. Two run orders from that set are displayed in Fig. 7. Finally, the third and fourth set contain run orders that are D-optimal for ρ values between 0.41 and 0.71 and above 0.71, respectively. Two run orders from each of these sets are displayed in Figs. 8 and 9. An alternative representation of the D-optimal run orders in Figs. 6–9 is given in Table 7.

Even more consistently than in the optimal run orders for the GLS estimator, sequences of axial and factorial runs can be observed in each of the optimal run orders for the OLS estimator. The six axial points are clustered in every optimal run order for every ρ value. Again, a given axial point is never preceded or succeeded by its mirror image. The factorial points of the central composite design appear in two clusters of four for any value of ρ. The points in each of these clusters are the same in every D-optimal run order found. The coordinates of the points in one cluster satisfy $x_1x_2x_3 = -1$, while those of the points in the second cluster satisfy $x_1x_2x_3 = +1$.

When $\rho \leqslant 0.41$, the optimal run orders for the OLS estimator start and end with a centre point, just like all optimal run orders in case the GLS estimator is used. However, for ρ values larger than 0.41, the optimal run orders either start or end with a centre run.

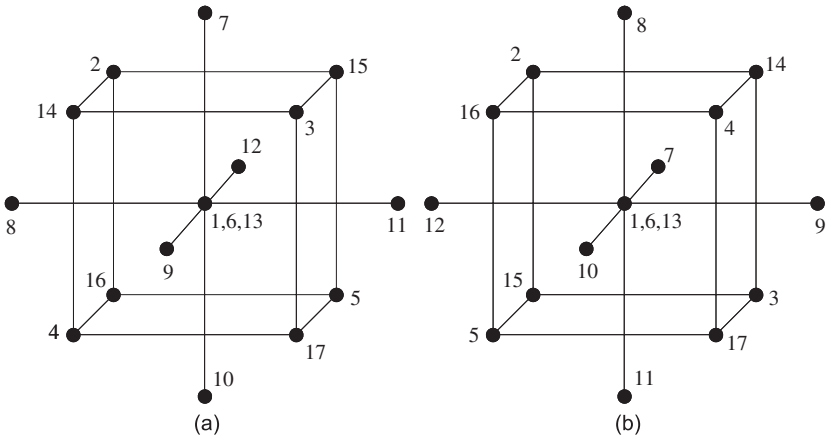

Fig. 8. D-optimal run orders for the OLS estimator when $0.41 \leqslant \rho \leqslant 0.71$.

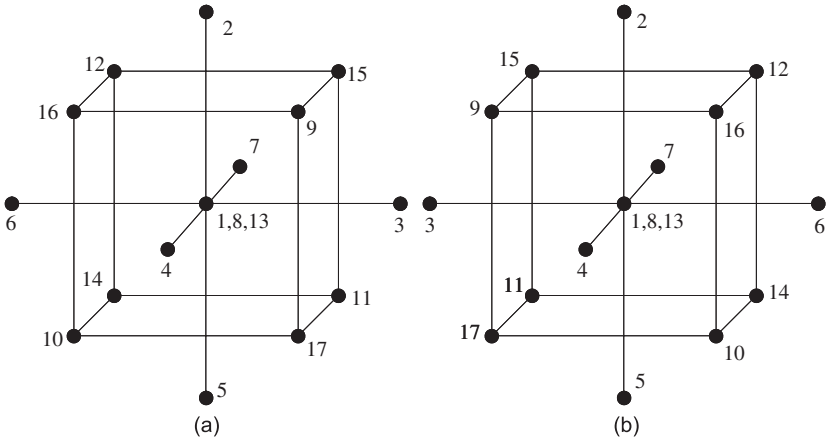

Fig. 9. D-optimal run orders for the OLS estimator when $0.71 \leqslant \rho < 1$.

Table 7
D-optimal run orders for the OLS estimator

ρ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0.0–0.11	15	1	4	6	7	15	11	14	10	12	13	9	8	5	3	2	15
	15	6	4	7	1	15	10	12	13	9	11	14	3	5	2	8	15
0.11–0.41	15	7	6	4	1	15	2	3	5	8	9	11	14	10	12	13	15
	15	8	5	3	2	15	1	4	6	7	9	11	13	10	12	14	15
0.41–0.71	15	4	7	1	6	15	12	9	13	11	10	14	15	3	8	2	5
	15	4	6	7	1	15	14	12	10	13	11	9	15	8	2	3	5
0.71–0.99	15	12	10	13	11	9	14	15	7	1	6	4	15	2	8	3	5
	15	12	9	13	11	10	14	15	3	5	2	8	15	6	4	7	1

The factorial, axial and centre points have labels 1–8, 9–14 and 15, respectively.

5.3. Random versus optimal run orders

In Section 3.2, it was shown that the order in which a central composite design is run has a substantial impact on the precision of the estimates. This was done by calculating the losses in D-efficiency of each generated random run order with respect to the best random run order found. We can now compute the losses in D-efficiency for each of the random run orders with respect to the D-optimal ones. The resulting side-by-side box-plots are given in Fig. 10. The Box-plots clearly show that the D-optimal

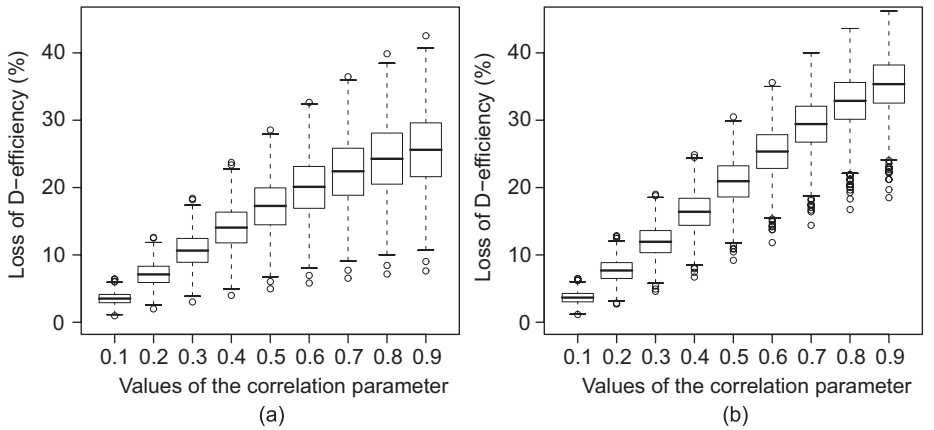

Fig. 10. Loss of D-efficiency when random instead of D-optimal run orders are used: (a) GLS; (b) OLS.

Table 8
D_{GLS}-efficiencies for the three sets of optimal run orders in Table 4 for various values of ρ

ρ	Set 1 (optimal for $\rho \leqslant 0.39$) (%)	Set 2 (optimal for $0.39 \leqslant \rho \leqslant 0.72$) (%)	Set 3 (optimal for $0.72 \leqslant \rho$) (%)
0.1	100.00	99.91	99.74
0.2	100.00	99.90	99.64
0.3	100.00	99.94	99.65
0.4	99.99	100.00	99.75
0.5	99.91	100.00	99.81
0.6	99.83	100.00	99.90
0.7	99.77	100.00	99.99
0.8	99.66	99.93	100.00
0.9	99.57	99.86	100.00

run orders are substantially better than the best of all random run orders that were generated, for all values of the correlation parameter. This demonstrates the usefulness of the variable-neighbourhood search algorithm.

As running the algorithm requires more time than generating a number of random run orders, researchers could be tempted to generate a large number of random run orders and select the one with the best D-criterion value. However, this does not lead to highly efficient run orders. For the 17-run central composite design, for example, generating and evaluating 1000 random run orders took about 0.5 s. Running the variable-neighbourhood search algorithm for that time yielded run orders that were more than 90% efficient for $\rho = 0.9$.

5.4. Robustness with respect to ρ

Prior to an experiment with serially correlated observations, researchers are unaware of the exact magnitude of the correlation parameter ρ. It is therefore useful to investigate the performance of an optimal run order for ρ values other than the one for which the run order was computed. Table 8 shows the D_{GLS}-efficiencies for the three sets of optimal run orders in Table 4 for various values of ρ. The table shows that run orders in Set 1 are optimal for small values of ρ and very nearly optimal for all other values of that parameter. Similar observations can be made for Sets 2 and 3. Consequently, when GLS estimation is used, the fact that ρ is unknown is not problematic because a run order that is optimal for one value of ρ is optimal or very nearly optimal for other values.

Table 9 shows the D_{OLS}-efficiencies for the four sets of optimal run orders in Table 7 for various values of ρ. The table shows that run orders in Sets 3 and 4 are optimal for large values of ρ and nearly optimal for small values of that parameter. The efficiencies of the run orders in Sets 1 and 2 drop to 97.63% and 98.83%, respectively, when ρ is large. These drops are larger than those in Table 8, but they are much smaller than the losses in efficiency incurred by using random run orders (these losses are shown in Fig. 10). Consequently, also when OLS estimation is used, the unknown nature of ρ is not problematic because a run order that is optimal for one value of ρ is optimal or nearly optimal for other values. The results suggest that it is better to overspecify ρ when computing an optimal run order than to underspecify it.

Table 9
D_{OLS}-efficiencies for the four sets of optimal run orders in Table 7 for various values of ρ

ρ	Set 1 (optimal for $\rho \leqslant 0.11$) (%)	Set 2 (optimal for $0.11 \leqslant \rho \leqslant 0.41$) (%)	Set 3 (optimal for $0.41 \leqslant \rho \leqslant 0.71$) (%)	Set 4 (optimal for $0.71 \leqslant \rho$) (%)
0.1	100.00	100.00	99.89	99.87
0.2	99.99	100.00	99.82	99.78
0.3	99.94	100.00	99.84	99.77
0.4	99.84	100.00	99.97	99.89
0.5	99.46	99.77	100.00	99.91
0.6	98.90	99.41	100.00	99.94
0.7	98.30	99.04	100.00	99.99
0.8	97.77	98.76	99.92	100.00
0.9	97.63	98.83	99.84	100.00

Table 10
Correlation matrix of the parameter estimates for the first D-optimal run order in Table 4 assuming $\rho = 0.3$

	I	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	x_1^2	x_2^2	x_3^2
I	1.00	0.06	0.04	−0.04	0.06	−0.04	0.05	−0.60	−0.64	−0.63
x_1	0.06	1.00	0.11	0.01	0.01	0.02	0.00	−0.09	−0.08	0.01
x_2	0.04	0.11	1.00	−0.08	0.11	0.18	0.00	−0.03	−0.03	−0.06
x_3	−0.04	0.01	−0.08	1.00	−0.16	0.04	0.04	0.06	0.00	0.05
x_1x_2	0.06	0.01	0.11	−0.16	1.00	0.01	0.00	−0.04	−0.04	−0.09
x_1x_3	−0.04	0.02	0.18	0.04	0.01	1.00	0.00	0.02	0.03	0.07
x_2x_3	0.05	0.00	0.00	0.04	0.00	0.00	1.00	−0.02	−0.03	−0.08
x_1^2	−0.60	−0.09	−0.03	0.06	−0.04	0.02	−0.02	1.00	0.38	0.32
x_2^2	−0.64	−0.08	−0.03	0.00	−0.04	0.03	−0.03	0.38	1.00	0.42
x_3^2	−0.63	0.01	−0.06	0.05	−0.09	0.07	−0.08	0.32	0.42	1.00

5.5. OLS versus GLS

As practitioners who are unaware of the presence of serial correlation typically use OLS, it is interesting to compare the optimal run orders for the two estimation methods. To this end, we computed the relative D-efficiency

$$D_{\mathrm{OLS/GLS}} = \left\{ \frac{|\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}(\mathbf{X}'_{\mathrm{OLS}}\mathbf{V}\mathbf{X}_{\mathrm{OLS}})^{-1}\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}|}{|\mathbf{X}'_{\mathrm{GLS}}\mathbf{V}^{-1}\mathbf{X}_{\mathrm{GLS}}|} \right\}^{1/p} \tag{3}$$

and the loss in D-efficiency, $(1 - D_{\mathrm{OLS/GLS}}) \times 100\%$, for different values of ρ. In these expressions, the model matrices $\mathbf{X}_{\mathrm{OLS}}$ and $\mathbf{X}_{\mathrm{GLS}}$ contain the optimal run orders for OLS and GLS estimation, respectively. It turns out that the loss in D-efficiency from using OLS instead of GLS estimation ranges from 0.50% when $\rho = 0.1$ to 28.08% when $\rho = 0.9$. OLS estimation is therefore not a good idea for large values of ρ, even not when an optimal run order is used.

The optimal run orders for OLS estimation are not always highly efficient when the data are analysed using GLS. This can be verified by evaluating the relative D-efficiency

$$\left\{ \frac{|\mathbf{X}'_{\mathrm{OLS}}\mathbf{V}^{-1}\mathbf{X}_{\mathrm{OLS}}|}{|\mathbf{X}'_{\mathrm{GLS}}\mathbf{V}^{-1}\mathbf{X}_{\mathrm{GLS}}|} \right\}^{1/p}$$

and the corresponding loss in D-efficiency. For example, the run orders in Figs. 8 and 9 lead to an efficiency loss of more than 6% when $\rho = 0.9$. By computing the relative D-efficiency

$$\left\{ \frac{|\mathbf{X}'_{\mathrm{GLS}}\mathbf{X}_{\mathrm{GLS}}(\mathbf{X}'_{\mathrm{GLS}}\mathbf{V}\mathbf{X}_{\mathrm{GLS}})^{-1}\mathbf{X}'_{\mathrm{GLS}}\mathbf{X}_{\mathrm{GLS}}|}{|\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}(\mathbf{X}'_{\mathrm{OLS}}\mathbf{V}\mathbf{X}_{\mathrm{OLS}})^{-1}\mathbf{X}'_{\mathrm{OLS}}\mathbf{X}_{\mathrm{OLS}}|} \right\}^{1/p},$$

it can be verified that run orders that are optimal for the GLS estimator can lead to a loss in D-efficiency of more than 20% when used in combination with OLS estimation. This shows that the D-optimal run orders are not robust in the sense that they lead to efficient estimates even if a different estimation method is utilized than the one for which they were obtained.

5.6. Features of the covariance matrix of the estimators

It is well known that the covariance matrix of the parameter estimator for a central composite design is a sparse matrix when the responses are statistically independent, so that the main effects and two-factor interaction effects can be estimated independently from each other and from the intercept and the quadratic effects. In the presence of an AR(1) correlation pattern,

the covariance matrix is no longer sparse, so that the parameter estimates are no longer independent. It turns out, however, that the covariance, and thus also the correlation, is nearly zero for most pairs of parameters when D-optimal run orders are used. This is illustrated in Table 10, where the correlation matrix of the parameter estimates for the first D-optimal run order in Table 4 assuming $\rho = 0.3$ is displayed. If the correlations between the estimates for the quadratic terms and the intercept are ignored, then only two of the correlations are substantially larger than 0.1 in absolute value. These are correlations between the estimates of a main effect and an interaction effect. They amount to 0.18 and -0.16.

6. Discussion

A variable-neighbourhood search algorithm was proposed for finding optimal run orders for given experimental designs. In this article, this was demonstrated assuming an AR(1) autocorrelation pattern and using two versions of the D-optimality criterion and a popular class of second-order response surface designs, namely the class of central composite designs. As far as the factorial portion of the central composite designs is concerned, the D-optimal run orders produced by the algorithm exhibit features that are similar to the ones obtained by Constantine (1989), Cheng and Steinberg (1991), and Martin et al. (1998b): the optimal run orders require large Hamming distances between the rows of the model matrix for consecutive factorial runs. The importance of large numbers of changes in values between the elements of the model matrix corresponding to pairs of consecutive factorial points is however in contrast with the overall picture. This is because the clusters of factorial and axial points in the D-optimal run orders limit the Hamming distances between consecutive rows in the full model matrix. Nevertheless, the D-optimal run orders outperform ones which do have a maximum number of level changes.

Despite the fact that the number of level changes in the full model matrix is not maximized, there are quite a number of level changes in the columns corresponding to the main effects of the experimental factors. These optimal run orders are therefore especially valuable for experiments involving factors for which it is not too hard, time-consuming or costly to reset the levels independently (for more details about the problem of running experiments in the presence of hard-to-change factors, see, for example Ju and Lucas, 2002; Webb et al., 2004; Goos and Vandebroek, 2004; Ganju and Lucas, 2005 and the references therein).

The algorithm presented here will prove a useful tool in the future search for optimal run orders in the presence of serial correlation and time trends. An important aspect of future research in this context will be to adopt a Bayesian perspective. This is because the optimal run orders not only depend on the magnitude of the serial correlation, but also because they depend on the type of serial correlation. In situations where researchers are uncertain about the precise nature of the serial correlation, a Bayesian approach that takes into account a researcher's prior beliefs will produce compromise run orders that guarantee an efficient estimation of the model parameters whatever the autocorrelation pattern turns out to be. Such a Bayesian approach will also be needed to construct trend-robust run orders when the exact nature of a time-trend is unknown prior to the experiment.

Another important aspect of future research is to seek optimal run orders for optimality criteria other than the D-optimality criterion used in this paper. This is especially important for experimental design problems where the focus is on precise prediction rather than on precise estimation of the model.

Given the computational complexity of the problem under study, improving the run times for the algorithm is another important concern for future research. One of the most promising research directions in this respect is to seed the algorithm with a high-quality *initial solution* (instead of a random one, as the current version of the algorithm uses) as this will shorten the time-consuming initial search towards the first high-quality solution. As we have shown in Section 5, good run orders exhibit special structures and it is likely that by carefully studying these, excellent initial solutions can be built.

Acknowledgement

Dr. Sörensen is a post-doctoral research fellow of the Flemish Fund for Scientific Research (FWO), which also sponsored the research of Mr. Garroi.

References

Atkinson, A.C., Donev, A.N., 1992. Optimum Experimental Designs. Clarendon Press, Oxford, UK.
Avanthay, C., Hertz, A., Zufferey, N., 2003. A variable neighborhood search for graph coloring. European J. Oper. Res. 151, 379–388.
Caporossi, G., Hansen, P., 2004. Variable neighborhood search for extremal graphs. 5. Three ways to automate finding conjectures. Discrete Math. 276, 81–94.
Cheng, C.-S., 1990. Construction of run orders of factorial designs. In: Ghosh, S. (Ed.), Statistical Design and Analysis of Industrial Experiments. Marcel Dekker, New York, pp. 423–439.
Cheng, C.-S., Steinberg, D.M., 1991. Trend robust two-level factorial designs. Biometrika 78, 325–336.
Constantine, G.M., 1989. Robust designs for serially correlated observations. Biometrika 76, 241–251.
Fleszar, K., Hindi, K.S., 2004. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. European J. Oper. Res. 155, 402–413.
Ganju, J., Lucas, J.M., 2005. Randomized and random run order experiments. J. Statist. Plann. Inference 133, 199–210.
Goos, P., 2002. The Optimal Design of Blocked and Split-plot Experiments. Springer, New York.
Goos, P., Vandebroek, M., 2004. Outperforming completely randomized designs. J. Quality Technol. 36, 12–26.
Hamming, R.W., 1950. Error detecting and error correcting codes. Bell System Tech. J. 26, 147–160.
Ju, H.L., Lucas, J.M., 2002. L^k factorial experiments with hard-to-change and easy-to-change factors. J. Quality Technol. 34, 411–421.
Kytöjoki, J., Nuortio, T., Bräysy, O., Gendreau, M., 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. Comput. Oper. Res. 34, 2743–2757.

Martin, R.J., Eccleston, J.A., Jones, G., 1998a. Some results on multi-level factorial designs with dependent observations. J. Statist. Plann. Inference 73, 91–111.
Martin, R.J., Jones, G., Eccleston, J.A., 1998b. Some results on two-level factorial designs with dependent observations. J. Statist. Plann. Inference 66, 363–384.
Mladenović, N., Hansen, P., 1997. Variable neighbourhood decomposition search. Comput. Oper. Res. 24, 1097–1100.
Mladenović, N., Petrović, J., Kovačević-Vujčić, V., Čangalović, M., 2003. Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search. European J. Oper. Res. 151, 389–399.
Myers, R.H., Montgomery, D.C., 1995. Response Surface Methodology: Process and Product Optimization using Designed Experiments. Wiley, New York.
Tack, L., Vandebroek, M., 2001. (D_t, C)-optimal run orders. J. Statist. Plann. Inference 98, 293–310.
Tack, L., Vandebroek, M., 2002. Trend-resistant and cost-efficient block designs with fixed or random block effects. J. Quality Technol. 34, 422–436.
Tack, L., Vandebroek, M., 2003. Semiparametric exact optimal run orders. J. Quality Technol. 35, 168–183.
Tack, L., Vandebroek, M., 2004. Budget constrained run orders in optimum design. J. Statist. Plann. Inference 124, 231–249.
Webb, D., Lucas, J.M., Borkowski, J.J., 2004. Factorial experiments when factor levels are not necessarily reset. J. Quality Technol. 36, 1–11.
Zhou, J., 2001. A robust criterion for experimental designs for serially correlated observations. Technometrics 43, 462–467.