Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

An algorithm for blocking regular fractional factorial 2-level designs with clear two-factor interactions

Ulrike Grömping[a,*]

[a] *Beuth University of Applied Sciences, Berlin, Germany*

27 July 2020

Abstract

Regular fractional factorial designs with 2-level factors are among the most frequently used experimental plans. In many cases, designs should be blocked for dealing with inhomogeneity of experimental units. At the same time, the research question at hand may imply a focus on specified sets of two-factor interactions, while it is not justified to assume negligibility of other low order effects. An algorithm is provided for blocking a regular fraction into – possibly small – blocks while keeping specified two-factor interactions clear from confounding with main effects or other two-factor interactions. The proposed algorithm is implemented in the R package **FrF2** and combines an estimability algorithm by the author with an automated implementation of a recent proposal for blocking fractions by hand.

Keywords: Fractional factorial 2-level designs; Blocked designs; Clear effects

1 Introduction

Regular fractional factorial designs with 2-level factors are frequently used. If inhomogeneity of experimental units has to be accounted for, they are typically conducted in blocks. At the same time, the research question at hand may imply a focus on specified sets of two-factor interactions, while it is not justified to assume negligibility of other low order effects. For example, in experiments with control factors and noise factors, particular emphasis may be on the estimation of interactions between control factors and noise factors, while not assuming that interactions within each group of factors are negligible. Two-factor interactions (2fis) that are not confounded with main effects or other 2fis are called "clear".

Blocking fractional factorial 2-level designs can be a challenging task, if the blocked design must keep a user-specified set of 2fis clear. Some authors have worked on this: For example, Chen et al. (2006) provided upper and lower bounds for the number of clear 2fis and proposed a construction method for blocked designs with many clear 2fis. Zhao, Li and Liu (2013) proposed an algorithm for constructing blocked resolution IV designs with maximum number of clear 2fis and provided a catalogue of such designs with up to 64 runs. A recent article by Godolphin (2019) has opened an interesting applied approach for this problem. Godolphin described her approach as "practitioner led" and provided catalogues of blocking templates for blocks of size 4 that practitioners can use for creating a tailor-made design that meets their needs for keeping specific 2fis clear. This paper provides an algorithm for the automation of her proposal, so that a design can be obtained from specifying estimability requirements, the number of experimental runs and the number of blocks. The algorithm is implemented in the R package **FrF2** (see Grömping 2014a for an earlier version of that package). Accommodation of estimability requirements – for blocked or unblocked designs – uses the estimability algorithm that was described in Grömping (2012), which makes use of the R package **igraph** (Csardi and Nepusz 2006) for subgraph isomorphism checking. Godolphin's perspective on blocking is transformed to that of the package **FrF2** (see Section 2), in order to combine it with the aforementioned estimability algorithm. The justification of Godolphin's blocking method is closely related to graph colouring and multipartite graphs; while the algorithmic implementation does not make use of methods related to graph colouring, these are

* Correspondence to: Ulrike Grömping, Dep. II, Beuth University of Applied Sciences, Luxemburger Str. 10, D-13353 Berlin, Germany. Fax: +49 (0) 30 4504 66 5127. E-mail address: groemping@beuth-hochschule.de.

+ This paper also includes online supplementary material providing fractions, code for examples and an additional example.

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

nevertheless important for competent use of the methodology and are therefore briefly recapitulated in Section 2.

Section 2 of this paper provides basic facts about regular fractional factorial designs, clear 2fis, blocking, graph theory and basic features of the R package **FrF2**. Section 3 explains the mathematical underpinnings for the algorithm of this paper and presents illustrative examples. Section 4 describes the algorithm and its implementation in the package **FrF2**. The final discussion points out limitations and opportunities for future extensions. Supplementary material provides details on the fractions used in this paper, an overview of relevant R functions in the R package **FrF2**, R code for the examples from Sections 2 and 3 and an additional worked example that accommodates user-specified custom generators.

2 Notation and basic facts

Regular capital letters (excluding the letter "I") denote experimental factors. Matrices are denoted by boldface capital letters, the superscript $\top$ denotes a transpose. All matrices in this paper have elements from the Galois field for the prime 2 (GF(2)={0,1}), and matrix multiplication uses modular arithmetic with modulus 2.

2.1 Regular fractions for 2-level factors

n treatment factors, each with levels 0 and 1 from GF(2), are to be investigated. A full factorial would consist of 2^n level combinations. A regular fraction in $N = 2^k = 2^{n-p}$ level combinations (i.e., a 2^{n-p} fraction) can be obtained by specifying $p > 0$ defining contrasts, which declare how p factors can be added to a full factorial in $k = n - p$ basic factors. For example, E=ABC and F=ABD can be used as defining contrasts for accommodating $p = 2$ additional factors in a full factorial of $k = 4$ factors A to D; this means that the levels of factors E and F are determined as the sums (modulo 2) of the levels of factors A, B and C or A, B and D, respectively.

In line with Godolphin (2019), the p defining contrasts will also be denoted via a $p \times (n - p)$ confounding matrix $\mathbf{Z}$, with ones for the basic factors involved in the defining contrast and zeroes for the others. Thus, for the design of Table 1 we have

$$\mathbf{Z} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \tag{1}$$

(columns labelled with letters A to D, rows with letters E and F).

A "word" is a factor combination whose sum is an $N \times 1$ constant vector with the neutral element 0, denoted as I. Defining contrasts imply defining words (here: ABCE, ABDF), and the p defining words span a word group of size 2^p (including the trivial word I itself) which can be used to work out the entire confounding structure of the fraction. See for example Table 1 for the confounding structure created from the defining contrasts E=ABC, F=ABD: the 2^2 words of the word group can be found in the first column of the table; furthermore, for each effect in the header, the table shows the confounded effects that can be deduced from the word group (as cosets).

An important quality criterion for a fraction is the word length pattern (WLP), which is the frequency table of lengths of non-trivial words in the word group (three words of length 4 in Table 1). The length of the shortest non-trivial word is called the resolution of the fraction and denoted as a Roman numeral (resolution IV in Table 1). The "minimum aberration" (MA) criterion ranks fractions by their WLPs: the fewer short words, the better; the fraction of Table 1 is optimal according to the MA criterion.

Fractions are called isomorphic, if their confounding structures are equivalent up to factor labelling. Catalogues of nonisomorphic fractions are available, see e.g. Chen, Sun and Wu (1993), Xu (2009), Block and Mee (2005, 2006) and Ryan and Bulutoglu (2010). These have been implemented in the R packages **FrF2** and **FrF2.catlg128**. Catalogued fractions are denoted as n–p.rank, where rank is the rank number in terms of

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

Table 1: The confounding structure from defining contrasts E=ABC and F=ABD. Headers give basic factor contrasts in Yates order, and the corresponding Yates column numbers. Bold face: main effects and 2fis (in breach of this paper's notational convention, bold face capitals in this table do not denote matrices).

Column	1	2	3	4	5	6	7	8
I	**A**	**B**	**AB**	**C**	**AC**	**BC**	ABC	**D**
ABCE ABDF CDEF	BCE BDF ACDEF	ACE ADF BCDEF	**CE** **DF** ABCDEF	ABE ABCDF DEF	**BE** BCDF ADEF	**AE** ACDF BDEF	**E** CDF ABDEF	ABCDE ABF CEF

Column	9	10	11	12	13	14	15
I	**AD**	**BD**	ABD	**CD**	ACD	BCD	ABCD
ABCE ABDF CDEF	BCDE **BF** ACEF	ACDE **AF** BCEF	CDE **F** ABCEF	ABDE ABCF **EF**	BDE BCF AEF	ADE ACF BEF	**DE** **CF** ABEF

the MA criterion (arbitrary in the case of ties), i.e. rank 1 denotes the best fraction in terms of the MA criterion. Thus, the MA fraction of Table 1 is catalogued as 6–2.1.

This section has given a concise account on regular fractions for 2-level factors. For a more detailed exposition, readers are referred to the comprehensive book by Mee (2009) or to Grömping (2014a).

2.2 Clear two-factor interactions and clear interactions graphs

The concept of clear 2fis relies on the assumption that interactions of more than two factors are assumed to be negligible. At the same time, there is a particular interest in certain 2fis, without the willingness to assume negligibility of any other 2fis. It then makes sense to call a 2fi "clear", if it is not confounded with a main effect or another 2fi. If the resolution of a fraction is at least V, all 2fis are clear. Obviously, if there is a defining word ABCD, the 2fis AB, AC, AD, BC, BD, CD cannot be clear. In Table 1, we can inspect the fraction for clear 2fis by removing all entries of lengths larger than two, i.e. keeping only the bold face entries: there are no clear 2fis (as these would be alone in a column); all main effects are clear, however.

In the following, the term clear interactions graph (CIG) will be used for any graph that depicts factors as vertices with each clear 2fi depicted as an edge between the two respective vertices. A CIG can represent a requirement set of 2fis to be kept clear (requirement CIG), the clear 2fis of a catalogued fraction (estimability CIG) or the clear 2fis of a blocked design (also called estimability CIG; the distinction between types of estimability CIG should be apparent from the context; in blocked designs, clear effects must not be confounded with the block factor). The following wordings will be used: the requirement set of clear 2fis will be identified with its CIG. It will be said that a design or fraction "can accommodate the requirement CIG" or "keeps the requirement CIG clear" in order to express that the fraction or design does not confound main effects or any 2fis from the requirement CIG with any other main effect, 2fi or block factor main effect (see also Section 2.4). For a fraction or design to keep the requirement CIG clear, its estimability CIG must contain the requirement CIG as a subgraph (possibly after vertex permutations, as usual in subgraph isomorphism checks).

Figure 1 shows a requirement CIG with the 10 clear 2fis AB, AC, BC, BD, BE, CD, CF, CG, EF, EG (the set $\mathcal{S}_2$ used by Godolphin 2019). The two different vertex colourings of the figure will be referred to later.

2.3 Elements of graph theory

This section introduces terminology for graphs, which will be used for dealing with properties of the CIGs. A graph consists of $n > 0$ vertices some or all of which are connected by edges; it is conceivable that a graph has no edges, e.g. the estimability CIG of Table 1. In this paper, the edges do not have a direction, i.e. they are undirected; furthermore, there is at most one edge between two vertices, and there are no edges from a

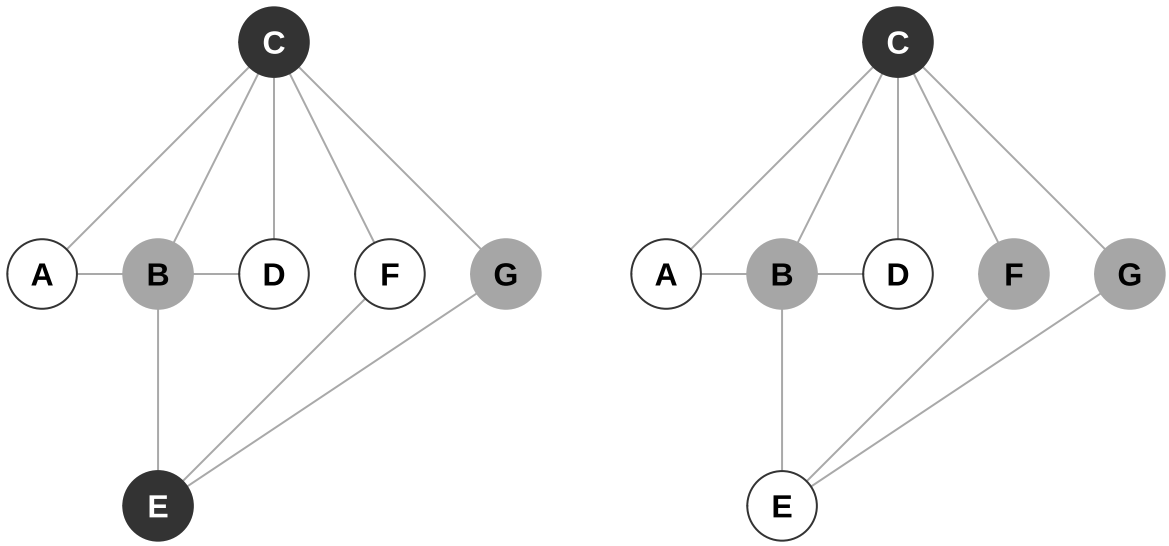

Figure 1: A requirement CIG with two different colourings.

vertex to itself. We thus consider so-called simple graphs only. Some terminology is now briefly explained. For background information, see for example Bondy and Murty (1976).

- A graph H is a *subgraph* of a graph G, if the vertices of H are a subset of the vertices of G, and the edges of H are a subset of the edges of G. The subgraphs considered in this paper are *spanning* subgraphs, which means that graph and subgraph have the same set of vertices.
- If two vertices are connected by an edge, they are called *neighbours*.
- A *clique* is a subgraph in which all vertices are pairwise neighbours.
- An *independent set* is a subgraph which has no edges. It is identified with the set of its vertices.
- For a graph G, its *complement* G^c denotes the graph for which vertices are neighbours if and only if they are not neighbours in G.
- Obviously, a clique in G is an independent set in G^c, and vice versa.
- The intersection of two graphs G_1 and G_2, each with vertices 1 to n, is the graph with all n vertices and those edges that are present in both graphs.
- The vertices of a *bipartite* graph can be partitioned into two independent sets. Analogously, the vertices of an *r-partite graph* can be partitioned into r independent sets. Edges only occur between the independent sets.
- For a *complete r-partite graph*, all edges between members of different independent sets are present.
- A partition of an r-partite graph into r non-empty independent sets $V_1, \ldots, V_r$ gives rise to a length r *r-profile* $P =< n_1, \ldots, n_r >$, with $n_1 \geq \cdots \geq n_r > 0$ the cardinalities of the independent sets. The r independent sets are also called *parts*.
- The r-profile of a complete r-partite graph is uniquely determined.
- *Proper graph colouring* refers to colouring the vertices of a graph such that no pair of neighbours has the same colour (see e.g. Figure 1 for two different proper colourings of the same graph).
- A graph is called *r-colourable* if it can be properly coloured with r colours. Of course, if a graph is r-colourable, it is also c-colourable, $r < c \leq n$.
- The *chromatic number* $\chi(G)$ of a graph G is the minimum over the r-values for which G is properly r-colourable, and a graph with chromatic number r is called r-chromatic.
- An r-partite graph is always r-colourable. The reverse is also true: any r-colourable graph is also r-partite. The graph in Figure 1 is thus both 3-colourable and 3-partite.
- Two graphs are called *isomorphic*, if they can be obtained from each other by relabelling of vertices. A graph G_2 is isomorphic to a *subgraph* of a graph G_1, if a relabelling of its vertices can be found such that all vertices and edges of G_2 are contained in G_1.

- If graphs G_1 and G_2 have n vertices each, graph G_1 is a complete r-partite graph, and graph G_2 is isomorphic to a subgraph of G_1, G_2 can be partitioned into independent sets according to the same r-profile as G_1 (this is a necessary condition for a subgraph relation).

For some r-colourable graphs, there is exactly one partition into r non-empty independent vertex sets, for example for complete r-partite graphs. In other cases, there are various possibilities for such partitions: for example, the 3-partite graph of Figure 1 has various possible colourings, two of which are shown in the figure and correspond to partitions {{A,D,F}, {B,G}, {C,E}} or {{A, D, E}, {B, F, G}, {C}}. The 3-profiles of these two partitions are <3,2,2> and <3,3,1>, respectively. It can be useful to identify partitions in order to support the search for a blocking that enables estimation of the required 2fis (see Examples 4 and 5, where Figure 1 will be re-visited).

Automated searching for all possible partitions of an r-partite graph is a difficult task, and a reliable implementation is not available, at least not in R software. For small graphs or graphs with a very regular structure, a human brain can often work something out from the graph's visualization. For example, in Figure 1, it is straightforward to see that the left-hand side representation would also permit various other colourings of the vertices in the middle row, as long as the dark colour is avoided and B remains different from A and D; for the right-hand side representation, the different colourings of vertices C and E imply a unique colouring of the remaining vertices.

The automated algorithm of this paper relies on subgraph isomorphism checking, which is another algorithmically difficult task. It uses the LAD algorithm (Solnon 2010) or the VF2 algorithm (Cordella et al. 2001) as implemented in the R package **igraph** (Csardi and Nepusz 2006). Use of these algorithms for accommodating a requirement CIG in an unblocked regular fractional factorial was described in Grömping (2012) and refined in Grömping (2014a).

2.4 Blocking designs

Designs are blocked in order to control for variation in the experimental material. Regular fractional factorial 2-level designs can be blocked into 2^{k-q} blocks of size 2^q. Where experimentation uses different batches of material (for example), a small number of large blocks may be sufficient, e.g. two blocks of size $N/2$ ($q = k - 1$) or four blocks of size $N/4$ ($q = k - 2$). However, it can happen that only smaller blocks can be made homogeneous enough, for example in the case of presenting physical samples to consumers in a choice set, where individual consumers should be treated as blocks and block sizes should not exceed eight so that the block factor must have (at least) $N/8$ levels.

It is customary to assume that

- the estimates for block factor effects are not themselves of interest,
- block factors do not interact with treatment factors,
- block effects are active so that treatment effects that are confounded with a block effect cannot be estimated.

The last bullet is the reason why 2fis in a blocked design are clear only if they are not confounded with blocks.

In line with Godolphin (2019), this paper looks at only a single block factor. It is possible to consider structured blocking, e.g. into Day and Shift per Day. This can be of interest for inspecting block variability but is not relevant for estimating treatment effects. Thus, a situation with four days and two shifts per day would be handled as a single block factor with eight blocks.

2.4.1 The two approaches for blocking

There are two different approaches for blocking a 2^{n-p} fraction into $2^{n-p-q} = 2^{k-q}$ blocks of size 2^q. These are initially explained for blocking the full factorial in factors A to D of Table 2 into four blocks of size 4 (i.e., $p = 0$, $n = k = 4$, $q = 2$). We will later refer back to the fraction of Table 1.

1) In the approach taken by R package **FrF2**, one specifies $k - q$ block generating contrasts (also called block generators) in order to get a block factor with 2^{k-q} levels whose degrees of freedom consist of

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

Table 2: The four rows of the principal block for two different blockings of the full factorial in factors A, B, C and D. Bold face: a 2×4 **X** matrix for the blocking. The table shows columns for all factorial effects, corresponding exactly to the columns of Table 1.

Column	**1**	**2**	3	**4**	5	6	7	**8**	9	10	11	12	13	14	15
I	**A**	**B**	AB	**C**	AC	BC	ABC	**D**	AD	BD	ABD	CD	ACD	BCD	ABCD
blocking 1							b_1				b_2	b_3			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**1**	1	**1**	1	0	0	**1**	1	0	0	0	0	1	1
0	**1**	**0**	1	**1**	0	1	0	**1**	0	1	0	0	1	0	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
blocking 2			b_1		b_2	b_3									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**0**	**0**	0	**0**	0	0	0	**1**	1	1	1	1	1	1	1
0	**1**	**1**	0	**1**	0	0	1	**0**	1	1	0	1	0	0	1
0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

the block generators and all their interactions: blocking 1 in Table 2 uses b_1=ABC and b_2=ABD as block generators, which implies the third degree of freedom for the 4-level block factor to be b_3=CD. In comparison, blocking 2 in the table uses block generators b_1=AB and b_2=AC, which gives rise to the third degree of freedom b_3=BC.

2) In Godolphin's (2019) approach, one specifies a principal block of $2^q = 4$ rows as a subgroup of the total $N = 2^k = 16$ rows and obtains 2^{k-q} blocks of size 2^q as this group and all its cosets. In this approach, one usually considers only the columns for the factor main effects (e.g. those with bold face headers in Table 2). Blocking 1 and blocking 2 in Table 2 show two different possible choices of a principal block: these are (1), bcd, acd, ab for blocking 1 and (1), d, abc, abcd for blocking 2 in the usual short-hand notation, which denotes rows by a lower case letter for each factor with a "1" in the row and the all-zero row as (1). The group of the 2^q rows of the principal block can be spanned by any q independent element rows. Godolphin introduced a $q \times n$ matrix **X** for denoting these generating rows of the principal block. For example, in Table 2 the bold-face elements indicate suitable choices for the two 2×4 **X** matrices. The group of rows for the principal block is then obtained from **X** by noting that
   - (1) is always in the group,
   - the rows of **X** themselves are in the group.
   - all sums of two or more rows of **X** are in the group.

   Constructing blocks in terms of Godolphin's approach is useful especially for small block sizes, because these correspond to small values of q and (relatively) large values of $k - q$.

Table 2 illustrates that the two approaches for constructing blocks are equivalent and can be used for constructing exactly the same block structures: the block degrees of freedom $b_1, \ldots, b_{2^{k-q}-1}$ of approach 1 are exactly those effects that are constant over all rows of the principal block in approach 2. In a full factorial, approach 1 produces the 2^{n-q} blocks from all level combinations of the $n - q$ block generators $b_1, \ldots, b_{n-q}$ (in Table 2, all level combinations of b_1 and b_2). In approach 2, each such level combination is equivalent to adding a specific (non-unique) constant row to the principal block for obtaining a particular coset. For example, for blocking 1 of Table 2, adding row 0,1,0,1=bd corresponds to $b_1 = 1, b_2 = 0$ (and $b_3 = 1$), adding row 0,0,0,1=d corresponds to $b_1 = 0, b_2 = 1$ (and $b_3 = 1$) and adding row 0,1,0,0=b corresponds to $b_1 = 1, b_2 = 1$ (and $b_3 = 0$).

2.4.2 Godolphin's approach for full factorials

A valid $q \times n$ **X** matrix for Godolphin's approach must have row rank q (in order to span a principal block of size 2^q) and must consist of columns from a set $\mathcal{X}_q$, which is defined as the set of non-zero columns from $GF(2)^q$. For example, $\mathcal{X}_2 = \{(0,1)^\top, (1,0)^\top, (1,1)^\top\}$. The cardinality of $\mathcal{X}_q$ is $2^q - 1$, i.e. there are at most

$2^q - 1$ different columns in $\mathbf{X}$.

The following properties hold for a full factorial blocked with a suitable $\mathbf{X}$ matrix:

Lemma 2.1 (compiled from Godolphin)**.** *If a full factorial is blocked using the principal block generated by a $q \times n$ $\mathbf{X}$ matrix with full row rank and columns from $\mathcal{X}_q$, the following results hold:*

- $\mathbf{X}$ *generates 2^{n-q} blocks of size 2^q.*
- *All main effects are unconfounded with blocks.*
- *All 2fis of two factors that have the same $\mathbf{X}$ column are confounded with blocks.*
- *All the other 2fis are unconfounded with blocks.*

Note that this indeed holds for Table 2, where factors C and D have the same $\mathbf{X}$ columns in blocking 1, so that their interaction corresponds to a constant all-zero column in the principal block. Confounding of at least one 2fi cannot be avoided, because the cardinality of $\mathcal{X}_2$ is three and thus smaller than n; choice of the $\mathbf{X}$ matrix allows us to decide which 2fi will be confounded with blocks. Blocking 2 of Table 2 confounds many more 2fis: the $\mathbf{X}$ matrix uses only two of the three possible columns of $\mathcal{X}_2$ and confounds AB, AC and BC with blocks. One would usually prefer blocking 1 over blocking 2.

2.4.3 Godolphin's approach for fractional factorials

Most results from Lemma 2.1 carry over to a fraction $\mathcal{F}$. Before looking at this formally, Example 1 applies the blockings of Table 2 to the fraction of Table 1.

Example 1: The runs of the fraction of Table 1 can be conducted in the blocks constructed in Table 2. The resulting confounding can be documented by adding a row for block degrees of freedom to the header row of confounded effects in Table 1. We see that blocking 1 is unsuitable, because it confounds factors E and F (columns 7 and 11) with the block factor. Blocking 2, on the other hand, does not confound main effects with the block factor. Each blocking is fully determined by the $q \times (n-p)$ $\mathbf{X}$ matrix of Table 2, which is now denoted as $\mathbf{X}_\mathrm{I}$ and defines the principal block based on the basic factors. Whether or not it produces a suitable blocking, depends on the resulting entries in the columns for generated factors E and F, which can be calculated as $\mathbf{X}_\mathrm{II} = \mathbf{X}_\mathrm{I}\mathbf{Z}^\top$, with $\mathbf{Z}^\top$ as given in (1):

For blocking 2,

$$\mathbf{X}_\mathrm{I} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{X}_\mathrm{II} = \mathbf{X}_\mathrm{I}\mathbf{Z}^\top = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

yield the $q \times n = 2 \times 6$ matrix

$$\mathbf{X} = \left(\mathbf{X}_\mathrm{I} \vdots \mathbf{X}_\mathrm{II}\right) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

(remember that matrix multiplication is modulo 2). The columns of the $q \times (n-p) = 2 \times 4$ matrix $\mathbf{X}_\mathrm{I}$ are labelled with letters A to D (corresponding to basic factors in Tables 1 and 2), the columns of the corresponding $q \times p = 2 \times 2$ matrix $\mathbf{X}_\mathrm{II}$ refer to generated factors E and F. $\mathbf{X}_\mathrm{II}$ corresponds to columns 7 and 11 of the 2nd and 3rd rows of the principal block for blocking 2 in Table 2.

Lemma 2.1 states that the $\mathbf{X}$ from (2) achieves a blocking of a 64-run full factorial for factors A to F into sixteen blocks of size 4, confounding 2fis AB, AC, BC, AE, BE, CE and DF with blocks (i.e. these edges are absent in the estimability CIG of the blocked full factorial in all six factors), while not confounding any other 2fis with blocks. Corollary 2.1 below transfers Lemma 2.1 to fractions, except for the number of blocks, i.e. a blocked fraction confounds exactly the same 2fis with blocks as the corresponding full factorial, and may of course confound further 2fis with each other.

Corollary 2.1. *Consider a 2^{n-p} fraction $\mathcal{F}$ with a $p \times (n-p)$ confounding matrix $\mathbf{Z}$. Let $\mathbf{X} = (\mathbf{X}_I \vdots \mathbf{X}_{II})$ with $\mathbf{X}_{II} = \mathbf{X}_I\mathbf{Z}^\top$ fulfill the requirements of Lemma 2.1. Then, the results of Lemma 2.1 hold for the fraction, with the only change that $\mathbf{X}$ generates only 2^{k-q} blocks of size 2^q, $k = n - p$.*

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

Example 1, continued: According to Corollary 2.1, matrix $\mathbf{X}$ from (2) creates four blocks each of size four in the fraction of Table 1 and confounds the 2fis AB, AC, BC, AE, BE, CE and DF with blocks. Fractionation reduces the number of blocks from sixteen (in the full factorial) to four. Had we chosen the $\mathbf{X}_{\mathrm{I}}$ matrix of blocking 1 of Table 2, ascertaining compatibility with the confounding structure by multiplication with $\mathbf{Z}$ would have yielded $\mathbf{X}_{\mathrm{II}}$ as a 2×2 matrix of zeroes, which means that Corollary 2.1 would not be applicable because the requirements of Lemma 2.1 would be violated (main effects of E and F would be confounded with the block factor when blocking the 64-run full factorial –or the fraction –based on the resulting $\mathbf{X}$).

2.5 Basics of R package FrF2

All calculations for this paper have been done with Version 2.2-2 of the R package **FrF2**; an earlier version was described in Grömping (2014a). The function `FrF2` produces regular fractional factorial 2-level designs. Its most basic form is to specify `nruns=` $N = 2^k = 2^{n-p}$ and `nfactors=` n, which will yield an experimental design based on the MA fraction n–p.1. Design construction is based on a large catalogue (`catlg`) of nonisomorphic fractions, which are sorted from good to bad in terms of the MA criterion.

The function `FrF2` allows users to specify a set of required clear 2fis (argument `estimable`). The approach for accommodating this requirement CIG has been described in Grömping (2012, 2014b). The most important points can be summarized as follows:

- The catalogue `catlg` contains the estimability CIG for each fraction.
- If a requirement CIG is specified in `FrF2`, the catalogue of suitable candidate fractions is checked from best to worst, whether the requirement CIG is isomorphic to a subgraph of the estimability CIG of the respective fraction.
- The design is constructed from the first (and thus best) successful fraction.

The workhorse function behind this algorithm (`mapcalc`) will also be used for the algorithm proposed in this paper. It uses subgraph isomorphism search functionality from the package **igraph** (Csardi and Nepusz 2006).

The function `FrF2` also allows users to block designs; in the simplest case, users can ask for a number of blocks, and `FrF2` searches for a blocking that does not confound any 2fi. If that cannot be found, users can permit the confounding of 2fis (`alias.block.2fis=TRUE`), which will often succeed. Until version 1.7.x, the focus was on larger blocks only, and blocking could not be combined with the functionality for estimable 2fis described in the previous paragraph. Version 2 of **FrF2** introduced the Godolphin approach for small blocks, and with it the possibility to combine blocking with requiring clear 2fis.

3 Clear estimability of 2fis for blocked designs

Following directly from Corollary 2.1, Corollary 3.1 provides the estimability CIG for a blocked fraction:

Corollary 3.1. *Consider a 2^{n-p} fraction $\mathcal{F}$ with a $p \times (n-p)$ confounding matrix $\mathbf{Z}$. Let $\mathbf{X} = (\mathbf{X}_I \vdots \mathbf{X}_{II})$ with $\mathbf{X}_{II} = \mathbf{X}_I \mathbf{Z}^\top$ fulfill the requirements of Lemma 2.1. Then, the estimability CIG of the blocked fraction $\mathcal{F}$ is the intersection of the estimability CIG of the unblocked fraction $\mathcal{F}$ and the blocked 2^n full factorial.*

Each column of the matrix $\mathbf{X}$ corresponds to a treatment factor. $\mathbf{X}$ thus partitions the treatment factors into parts according to the element of $\mathcal{X}_q$ that is assigned to the factor: there are up to $2^q - 1$ non-empty parts. Lemma 2.1 implies that the estimability CIG of a full factorial blocked according to $\mathbf{X}$ is a complete r-partite graph with the parts induced by $\mathbf{X}$.

Example 1, continued: The estimability CIG of the 64-run full factorial blocked based on $\mathbf{X}$ from (2) is a bipartite graph with all edges between parts {A,B,C,E} and {D,F} present ($4 \cdot 2 = 8$ edges) and the seven within-part edges (AB, AC, AE, BC, BE, CE, DF) missing. The unblocked fraction has no clear 2fis, i.e. its estimability CIG does not have any edges. Thus, according to Corollary 3.1, the estimability CIG of the blocked fraction is also a graph without edges. The above-mentioned seven 2fis corresponding to

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

within-part edges are confounded with the block factor (columns 3, 5 and 6 of Table 1), whereas the eight 2fis corresponding to between-part edges are unconfounded with the block factor but are already confounded in four pairs in the unblocked fraction (columns 9, 10, 12 and 15 of Table 1).

In general, there are four types of 2fis:

- 2fis that are clear in the blocked fraction,
- 2fis that are clear in the unblocked fraction but confounded with blocks,
- 2fis that are unconfounded with blocks but not clear in the unblocked fraction and
- 2fis that are both confounded in the unblocked fraction and confounded with blocks.

The small Example 1 does not have any 2fis of the first two types. See Example 3 in Section 3.3 for a 2^{13-6} fraction that is blocked into blocks of size 8 for which all four types of 2fis occur among the overall 78 2fis: There, the blocked full factorial has 69 clear 2fis, the unblocked fraction has 66 clear 2fis. The blocked fraction has 65 clear 2fis (the first type, intersection of the two estimability CIGs), one 2fi is clear in the unblocked fraction but confounded with the block factor, four 2fis are unconfounded with blocks but confounded in the fraction, and eight 2fis are confounded both with blocks and in the unblocked fraction.

3.1 Profile and number of two-factor interactions unconfounded with blocks

The $\mathbf{X}$ matrix defines r parts for which the estimability CIG of the corresponding blocked full factorial is a complete r-partite graph. The r-profile of that CIG (see Section 2.3) will also be called the *profile* of $\mathbf{X}$. This profile directly implies the number of clear 2fis in the blocked 2^n full factorial, or equivalently, the number of 2fis *unconfounded with blocks* when blocking a 2^{n-p} fraction with this $\mathbf{X}$ (provided that $\mathbf{X}$ is suitable for blocking the fraction).

Godolphin (2019) provided a formula for the number of clear 2fis in a blocked full factorial based on the profile $< n_1, \ldots, n_{2^q-1} >$ (see end of her section 2): there are

$$\sum_{i=1}^{2^q-2} \sum_{j=i+1}^{2^q-1} n_i n_j \tag{3}$$

clear 2fis. This is straightforward to see, as (3) is the number of between-part edges in the full (2^q-1)-partite graph with part sizes n_1 to n_{2^q-1}. Godolphin (2019, Theorem 2) further provided a (sharp) upper bound for that number:

$$\phi_{\max} = \binom{n}{2} - vw - (2^q-1)\binom{v}{2}, \tag{4}$$

with $v = [n/(2^q-1)]$ and $w = n - (2^q-1)v$, where $[\cdot]$ denotes the floor function. This bound equals Formula (3) for the maximally balanced profile: if $w = 0$, all parts have size v, and $\phi_{\max}$ subtracts the number of (non-clear) within part edges from the total number of pairs. If $w > 0$, the most balanced setup consists of $2^q - 1 - w$ parts of size v and w parts of size $v+1$; for the latter, vw additional edges must be subtracted. $\phi_{\max}$ is attained if the largest and smallest frequencies of columns of $\mathcal{X}_q$ used in $\mathbf{X}$ differ by at most 1 (including zeroes for absent elements of $\mathcal{X}_q$).

Example 1, continued: The profile induced by the $\mathbf{X}$ matrix (2) is <4,2,0>, because of four instances of column $(0,1)^\top$, two instances of column $(1,0)^\top$ and zero instances of column $(1,1)^\top$. Zeroes are often omitted from the profile, i.e. the profile might also be specified as <4,2>. There are $4\cdot 2 + 4\cdot 0 + 2\cdot 0 = 4\cdot 2 = 8$ clear 2fis in the blocked 64-run full factorial; consequently, in the blocked fraction, these 8 are also unconfounded with blocks (but in this example confounded with other 2fis). As can be seen in this example, whether or not the profile contains trailing zeroes does not make a difference in calculating the number of 2fis that are unconfounded with blocks.

3.2 Search for the best X

For a full factorial, every $q \times n$ matrix $\mathbf{X}$ with columns from $\mathcal{X}_q$ and rank q can be used for blocking, i.e. every conceivable profile with up to $2^q - 1$ parts can be constructed. It is thus straightforward to handcraft a

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

suitable $\mathbf{X}$ matrix, for example with the most balanced profile for obtaining the maximum number of clear 2fis.

For blocking a particular *fractional* factorial with confounding matrix $\mathbf{Z}$, the structural requirement $\mathbf{X}_{\text{II}} = \mathbf{X}_{\text{I}}\mathbf{Z}^\top$ restricts possibilities. It may render some profiles infeasible, because there are no valid $\mathbf{X}$ matrices with these profiles, where conditions for validity were specified in Lemma 2.1. This can even happen for resolution V or higher fractions.

Example 2: A design in 256 runs with 13 factors is to be run in 64 blocks of size 4. The resolution V MA fraction 13–5.1 from the catalogue `catlg` has the generating contrasts J=ABCDEFG, K=ABCDH, L=ABEFH, M=ACEGH and N=ADFG. The $p \times (n-p) = 5 \times 8$ matrix $\mathbf{Z}$ is

$$\mathbf{Z} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{5}$$

A brute force search over all possible $\mathbf{X}_{\text{I}}$ matrices shows that only four valid profiles can be obtained for blocking this fraction into blocks of size 4: <5,5,3>, <7,3,3>, <7,5,1> and <9,3,1>. Because the fraction has resolution V, the estimability CIG of the blocked fraction equals the estimability CIG of the blocked full factorial, which is the complete 3-partite graph that results from the profile. Figure 2 visualizes the estimability CIGs for the four profiles: since they are complete 3-partite graphs, they contain all edges between vertices of different colours and no edges between vertices of the same colour.

In Example 2, a brute force search was used. Godolphin (2019) proposed to fix the first column of $\mathbf{X}_{\text{I}}$, and inspect all $(2^q-1)^{k-1}$ possible choices of $k-1$ columns from $\mathcal{X}_q$ for the remaining columns of $\mathbf{X}_{\text{I}}$. For each $\mathbf{X}_{\text{I}}$ so obtained, she proposed to obtain $\mathbf{X}_{\text{II}} = \mathbf{X}_{\text{I}}\mathbf{Z}^\top$, to discard the case if the row rank of $\mathbf{X}$ is less than q or an all-zero column is found, and to record the profile of the resulting valid $\mathbf{X}$ otherwise. In the worst case (impossible request), this search procedure has to check $(2^q-1)^{k-1}$ $\mathbf{X}$ matrices.

A more efficient brute force search is implemented in the package **FrF2**: since it does not matter which column from $\mathcal{X}_q$ is used for which colour, one can not only fix the first column, but can more generally restrict the cth column to at most c choices. For describing the procedure, denote $\mathcal{X}_q = \{\xi_1, \dots, \xi_{2^q-1}\}$ (it does not matter, *how* the elements of $\mathcal{X}_q$ are ordered, only *that* they are ordered). Choices for $\mathbf{X}_{\text{I}}$ are explored as follows:

- Fix the first column of $\mathbf{X}_{\text{I}}$ to be ξ_1 (one choice).
- If $2^q - 1 \geq 2$, pick the second column of $\mathbf{X}_{\text{I}}$ from $\{\xi_1, \xi_2\}$ (two choices); otherwise pick it from $\mathcal{X}_q$ ($2^1 - 1 = 1$ choice).
- ...
- If $2^q - 1 \geq c$, pick the cth column of $\mathbf{X}_{\text{I}}$ from $\{\xi_1, \dots, \xi_c\}$ (c choices); otherwise pick it from $\mathcal{X}_q$ ($2^q - 1$ choices).
- ...
- If $2^q - 1 \geq k = n - p$, pick the kth column of $\mathbf{X}_{\text{I}}$ from $\{\xi_1, \dots, \xi_k\}$ (k choices); otherwise pick it from $\mathcal{X}_q$ ($2^q - 1$ choices).

This approach reduces the worst-case number of $\mathbf{X}_{\text{I}}$ matrices to be checked to $k!$ for $k \leq 2^q - 1$ and to $(2^q-1)!(2^q-1)^{k+1-2^q}$ otherwise. Like for Godolphin's search procedure, choices with fewer than q different columns in $\mathbf{X}$ are discarded (violation of full row rank), as are choices with all-zero columns in $\mathbf{X}_{\text{II}}$. For $k \geq 2^q - 1$, the proportion of worst case matrices to be checked in comparison to Godolphin's search procedure is $(2^q-2)!(2^q-1)^{3-2^q}$, which is 1 for $q=1$, 2/3 for $q=2$, 0.0428 for $q=3$ and 0.00004 for $q=4$ (i.e. the larger q, the larger the gain by the modified approach).

Example 2, continued: For blocking the fraction 13–5.1 into blocks of size 4, at most $(2^q-1)!(2^q-1)^{k+1-2^q} = 3! \cdot 3^5 = 1458$ $\mathbf{X}_{\text{I}}$ matrices need to be inspected. The automated brute force search of the package **FrF2** for the best $\mathbf{X}$ used the elements of $\mathcal{X}_2$ in the order $\xi_1 = (0,1)^\top$, $\xi_2 = (1,0)^\top$ and $\xi_3 = (1,1)^\top$. For the first

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

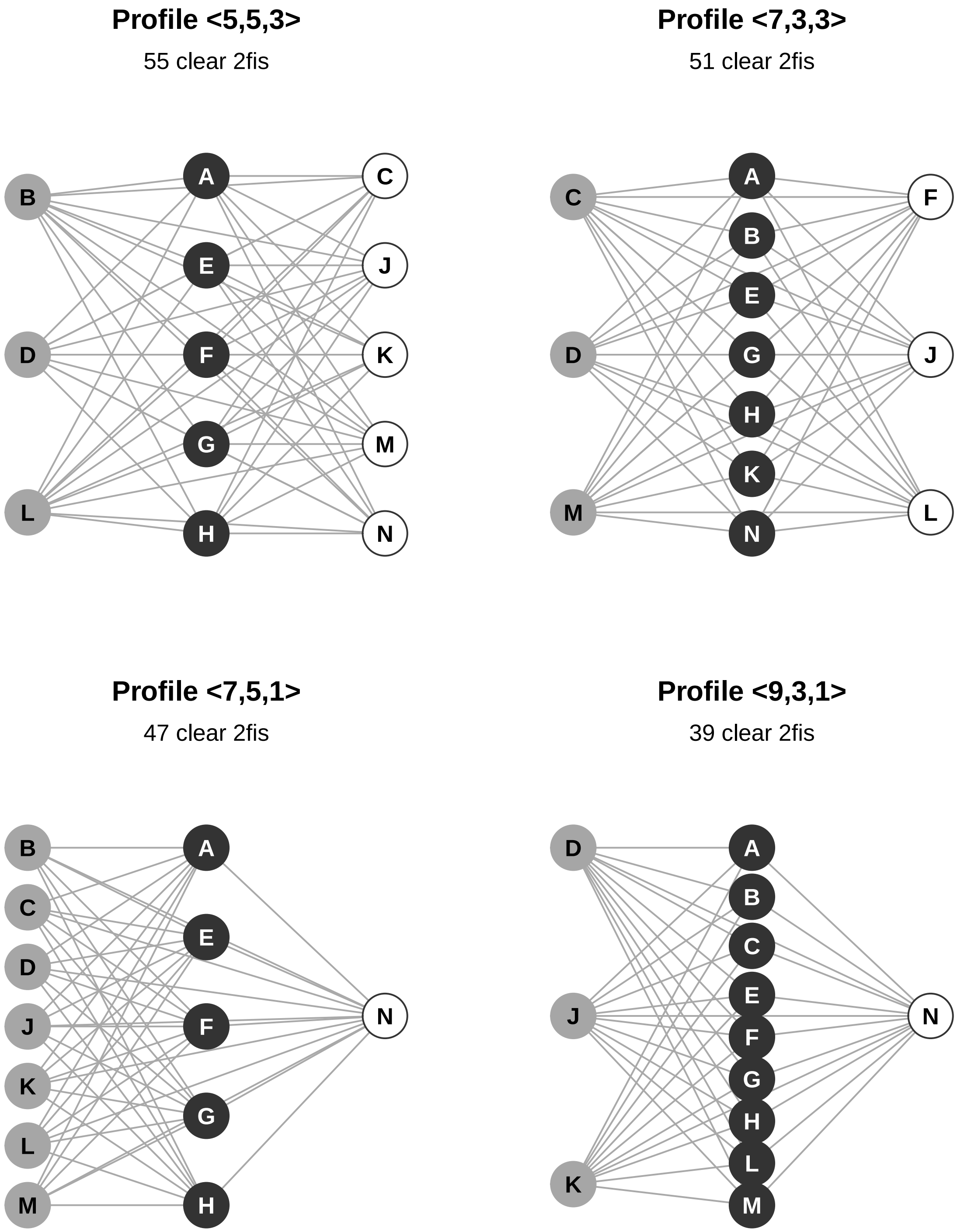

Figure 2: Estimability CIGs for the four different profiles of blocking the fraction 13-5.1 into blocks of size 4.

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

column of $\mathbf{X}$, ξ_1 was fixed; for the second column of $\mathbf{X}$, possible choices were ξ_1 and ξ_2; for the third and all further columns of $\mathbf{X}$, all elements of $\mathcal{X}_2$ were available. The search yielded

$$\mathbf{X} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{6}$$

within less than a second. The profile of this $\mathbf{X}$ is <5,5,3>, i.e. the most balanced one among the four possible profiles for the fraction. Note that the catalogue in the R package **FrF2** contains only the MA fraction 13–5.1 for 13 factors in 256 runs. Example 8 in Section D of the supplementary material provides a custom set of generating contrasts for a 2^{13-5} fraction for which a larger variety of profiles can be obtained, among them the most balanced profile <5,4,4>, which attains the maximum number of clear 2fis from (4).

For $q = 1$, the unique $\mathbf{X} = (1, \ldots, 1)$ (n elements) implies that all 2fis are confounded with the block effect. Consequently, all effects that involve an even number of factors are interactions of block degrees of freedom and thus also confounded with the block effect. This implies that fractions with any odd-length words cannot be used for blocking into blocks of size 2, because main effects of all factors from odd-length words would be confounded with the block factor. This observation will be revisited in Section 4.

3.3 Number of clear two-factor interactions

For blocking full factorials or fractions with resolution V or higher, the number of clear 2fis of the blocked design coincides with the number of 2fis that are unconfounded with blocks. For resolution IV fractions, it can be smaller, and the best overall result is not necessarily achieved by the $\mathbf{X}$ matrix that confounds the fewest 2fis with blocks.

Example 3: Blocking the resolution IV fraction 13–6.1 into blocks of size 8, the maximum number of achievable clear 2fis is 65. The automatically created design is obtained with

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{7}$$

(profile <4,2,2,2,1,1,1>) and has the 65 clear 2fis AC, AD, AE, AF, AG, AH, AJ, AK, AM, BC, BD, BE, BF, BG, BH, BJ, BK, BM, CE, CG, CH, CJ, CK, CL, CN, DE, DG, DH, DJ, DK, DL, DN, EF, EG, EH, EJ, EK, EL, EM, EN, FG, FH, FJ, FK, FL, FN, GH, GK, GL, GM, GN, HJ, HK, HL, HM, HN, JK, JL, JM, JN, KL, KM, KN, LM, MN. The full factorial blocked with this $\mathbf{X}$ matrix would keep 69 2fis clear. Twelve 2fis of the unblocked fraction are not clear, among them CD, CM, DF and FM, which would have been clear in a blocked full factorial with the $\mathbf{X}$ from (7). A single 2fi, GJ, would be clear in the unblocked fraction, but is confounded with blocks. Eight 2fis are confounded in the fraction and also confounded with blocks. The most balanced profile compatible with the fraction is <3,2,2,2,2,1,1>. Blocking a full factorial with this more balanced profile keeps 71 2fis clear; because of a smaller overlap between confounded 2fis, blocking the fraction with this profile keeps at most 61 2fis clear. It has been checked (see the code in the supplementary material) that the number of clear 2fis cannot be increased beyond 65 by using a different resolution IV fraction.

3.4 Keeping specific two-factor interactions clear

3.4.1 Full factorial case

Lemma 3.1 (Godolphin, Lemma 1)**.** *A requirement CIG can be accommodated in a full factorial design in blocks of size* 2^q *if and only if it is* $(2^q - 1)$*-colourable.*

This result follows directly from the identification of $\mathbf{X}$ matrix profiles with graph profiles. For accommodating a particular requirement CIG in the blocked design, the requirement CIG must be a subgraph of the estimability CIG of the blocked design, which is a complete r-partite graph with partitions according to the columns of the $\mathbf{X}$ matrix that was used for blocking. This implies that $\mathbf{X}$ must be chosen such that the parts of the blocked design correspond to valid parts of the requirement CIG. The construction of a suitable matrix **X** thus corresponds to a graph colouring exercise for the requirement CIG, with each colour corresponding to a particular element of $\mathcal{X}_q$, following the rules from Lemma 2.1. Contrary to classical graph colouring, for which one would attempt to colour a graph with as few colours as possible, the partition chosen for the requirement CIG should consist of as many parts as possible (i.e. ideally $2^q - 1$), even if a smaller number of parts would suffice, in order to achieve as many clear 2fis as possible. If the requirement CIG permits more than one partition with the maximum possible number of parts, the most balanced one should be chosen in order to maximize the number of clear 2fis in the blocked design.

Example 4: A full factorial design in seven factors is to be blocked into blocks of size 4, such that the requirement CIG of Figure 1 is kept clear. The two colourings in the figure correspond to profiles <3,2,2> and <3,3,1>, respectively. Since the profile <3,2,2> is maximally balanced, it is an ideal choice in terms of maximizing the number of clear 2fis. The 3-partition corresponding to the depicted profile is {{A,D,F},{B,G},{C,E}}. Using vectors $(0,1)^\top$, $(1,0)^\top$, and $(1,1)^\top$ in this order,

$$\mathbf{X} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{8}$$

The blocked design keeps 16 2fis clear, i.e., it confounds 5 2fis with the block main effect. The 10 2fis from the requirement CIG of Figure 1 are among the clear 2fis.

Godolphin formulated several sufficient conditions for a requirement CIG to be compatible with 2^{n-q} blocks of size 2^q; these are not repeated here.

3.4.2 Fractional factorial case

The following lemma assembles necessary conditions for blocking a fraction while keeping specified 2fis clear.

Lemma 3.2. *Let $\mathcal{F}$ denote a fraction of resolution at least IV that is to be blocked into blocks of size 2^q, while keeping a requirement CIG G clear. The following conditions are necessary:*

(i) $\mathcal{F}$ keeps G clear, i.e. the estimability CIG of $\mathcal{F}$ contains a subgraph that is isomorphic to G.
(ii) G is $(2^q - 1)$-colourable.
(iii) Blocking $\mathcal{F}$ into blocks of size 2^q permits a $2^q - 1$-profile that corresponds to a valid $(2^q - 1)$-profile of G.

Condition (i) is equivalent to Godolphin's C1. It implies that the algorithm may restrict the search to fractions that keep the requirement CIG clear. Because of Lemma 3.1, condition (ii) is necessary even for blocking a full factorial in n factors into blocks of size 2^q, which implies necessity for blocking $\mathcal{F}$. Condition (iii) follows from the fact that the requirement CIG must be isomorphic to a subgraph of the estimability CIG; it can only be checked by attempting the blocking.

According to Corollary 3.1, the estimability CIG of the blocked design is the intersection of the estimability CIGs of the blocked full factorial and of the unblocked fraction. If a fraction is to be blocked while accommodating a requirement CIG, we can regard this either as accommodating a requirement CIG in the estimability CIG from a blocked fractional factorial, or as blocking a fractional factorial that has been tailored to accommodate the requirement CIG.

The former approach was taken by Godolphin: she provided catalogues of templates for blocking fractions into blocks of size 4, by listing factor partitions corresponding to feasible **X** matrices. For resolution V fractions, her catalogues contain at most one entry for each profile. For resolution IV fractions, there may be more than one entry for listed profiles, and the templates also list the 2fis that are confounded in the fraction although they would be unconfounded in the blocked full factorial; however, not all templates are listed (e.g., there are no templates with "1" entries in the profile for larger settings). Based on the templates, it is the practitioner's task to find an appropriate allocation of experimental factors such that the requirement CIG is accommodated (which can be quite tedious, especially for the resolution IV case).

In this paper, the algorithm described in Grömping (2012) pre-filters fractions for compatibility with the requirement CIG. Blocking is attempted only for those pre-filtered fractions. For a valid candidate fraction, for every $\mathbf{X}$ matrix, a subgraph isomorphism check determines whether the requirement CIG can be accommodated in the estimability CIG of the blocked design. The entire process is automated, so that users are freed from the tedious and error-prone task of working out a suitable allocation of experimental factors. If the algorithm is unsuccessful for the candidate, a new search among catalogued fractions with worse aberration can be triggered.

Example 5: Like in Example 4, the requirement CIG of Figure 1 is to be kept clear. Instead of a full factorial with 128 runs, we now use the MA quarter fraction 7–2.1, blocked into eight blocks of size 4 ($q = 2$). The estimability CIG of the unblocked fraction is shown in Figure 3, coloured according to the successful mapping found by the algorithm; that mapping has profile <3,3,1>, corresponding to the right-hand side colouring in Figure 1. All the necessary conditions of Lemma 3.2 are satisfied; condition (iii) follows from the fact that a solution was found. (Note that an $\mathbf{X}$ matrix with profile <3,2,2> like in the left-hand side colouring of Figure 1 is also compatible with the fraction, but does not keep the requirement CIG clear; this underlines that condition (iii) of Lemma 3.2 is necessary, but not sufficient.) The algorithm implemented in **FrF2** returned the mapping A:1, B:4, C:7, D:2, E:5, F:3, G:6, together with an $\mathbf{X}$ matrix that corresponds to the partition {{A,D,E}, {B,F,G}, {C}} (see the right-hand side graph in Figure 1). With this partition and $\mathbf{X}$ columns ordered according to the map, any $\mathbf{X}_\mathrm{I}$ matrix (for requirement CIG columns A D F B E in this order) corresponding to the requirement CIG profile yields the entire $\mathbf{X}$ matrix for blocking the fraction through the combination with $\mathbf{X}_\mathrm{II} = \mathbf{X}_\mathrm{I}\mathbf{Z}^\top$ (modulo 2) (for factors G and C, in this order), as usual. The matrices are as follows:

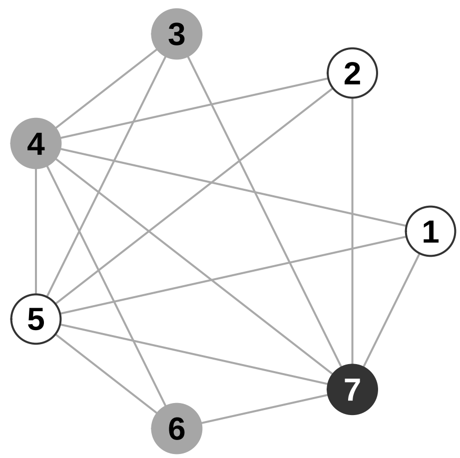

Figure 3: Estimability CIG of the unblocked MA fraction 7-2.1, coloured in terms of the successful mapping for Example 5 (this is of course no proper colouring for this CIG). The generating contrasts for the fraction are: factor 6 from the 3-factor interaction of factors 1, 2, and 3 and factor 7 from the 4-factor interaction of factors 1, 2, 4, and 5.

$$\mathbf{X}_\mathrm{I} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \qquad (9)$$

Note that the column order of the matrix $\mathbf{Z}$ is not remapped but corresponds to the original factor order in the fraction 7–2.1.

Example 6: Nine factors are to be accommodated in 64 runs with blocks of size 4. They consist of seven control factors C1 to C7 and two noise factors N1 and N2. All interactions of noise factors with control factors are to be kept clear. The requirement CIG is obviously 3-colourable (it is even 2-colourable, one colour for noise factors and one colour for control factors would suffice). The unblocked fraction 9–3.1 is able to accommodate the requirement CIG. It can also be suitably blocked. Note that a suitable blocking can be found only because a subgraph isomorphism check is done for each **X** matrix during the search process (in step 4 of the algorithm shown in Figure 5 of Section 4; demonstrated in the supplementary material).

Example 7: An experiment in 13 factors is to be conducted in 128 runs in blocks of size 4. All interactions with one particular factor are to be kept clear. The requirement CIG is 2-chromatic and thus of course 3-colourable (i.e. condition (ii) of Lemma 3.2 is fulfilled). It can be accommodated in various unblocked fractions, among them 13–6.1, 13–6.2, 13–6.3 and 13–6.4 (i.e. condition (i) of Lemma 3.2 is fulfilled for these fractions). For condition (iii) of Lemma 3.2, note that all valid partitions for the requirement CIG have to keep the particular factor in a part of its own, i.e. all valid profiles of the requirement CIG have to have at least one element that is "1". The fraction 13–6.3 has MA among fractions that fulfil the necessary condition (iii) of Lemma 3.2. With suitable choice of an **X** matrix, a blocked design based on fraction 13–6.3 keeps a total of 36 2fis clear. The maximum possible number for a full factorial blocked according to the most balanced profile with a singleton would be 48 ($6 \cdot 6 + 2 \cdot 6 \cdot 1$ from profile <6,6,1>). A search over all 197 blocked fractions whose estimability CIG contains the requirement CIG showed that 40 clear 2fis are possible from fraction 13–6.16. It is then the user's decision whether to prefer the fraction 13–6.3 with the better MA behaviour of the unblocked fraction or the fraction 13–6.16 which yields more clear 2fis in the blocked fraction.

Note that the successful fraction 13–6.3 of Example 7 is *dominated* and is therefore not available in the standard catalogue of the package **FrF2**; availability of catalogued fractions will be further discussed in Section 4.

4 The algorithm and its implementation in the R package FrF2

A resolution IV or higher fraction for n factors in $N = 2^{n-p} = 2^k$ runs is to be blocked into 2^{k-q} blocks of size 2^q, such that many 2fis are clear, or such that a requirement CIG can be accommodated. Figures 4 and 5 give a high-level overview of the algorithm implemented for this purpose: Figure 4 describes the selection of eligible candidate fractions; for full factorials, or where a requirement CIG is specified, only a single candidate is selected. The algorithm then loops over the selected candidates; for each candidate in the list, the algorithm of Figure 5 is applied. In the case of success for the current candidate, the search finishes successfully. Otherwise the algorithm switches to the next candidate fraction. Where a requirement CIG is specified and the single candidate was not successful, step 11 of Figure 5 can be followed with re-entering step 3 of Figure 4, eliminating the failed candidate from the eligible fractions. At present, this restart has to be manually triggered (see e.g. the code in the supplementary material for improvement attempts in Example 3).

The following subsections will shed light on the implementation and on relevant resource constraints for four different scenarios (full or fractional factorials, without or with a requirement CIG). Run times have been inspected on my Windows 10 machine with an Intel i7 2.4 GHz processor with four cores and 32GB RAM. The examples from Sections 2 and Section 3 have very short run times, usually lower than one second. Likewise, reproducing all scenarios from Tables 1 and 2 of Godolphin (2019; profiles for blocks of size four in resolution V or higher fractions of sizes 32 to 128 runs) took less than four seconds. However, even for a single unblocked full factorial or fraction, run times can be much longer if designs become very large and/or infeasibility of a request has to be confirmed by checking all possible **X** matrices.

4.1 Full factorial without a requirement CIG

The automatic algorithm will quickly return a blocked design with $\phi_{\max}$ clear 2fis. The first $\min(2^q - 1, n)$ factors will be placed in separate parts. For more than $2^q - 1$ factors, let us number parts as $1, \ldots, 2^q - 1$); factor i will be placed in part i modulo $2^q - 1$ (where 0 corresponds to part $2^q - 1$). Alternatively, the user can manually create a suitable **X** matrix with desired properties and create a design from it using the R function `FF_from_X`.

Failure to create a blocked design may be caused by resource problems: A suitable **X** matrix for creating blocks of size 2^q ($1 \le q < n$) is found very fast even for quite large numbers of factors; for example, for

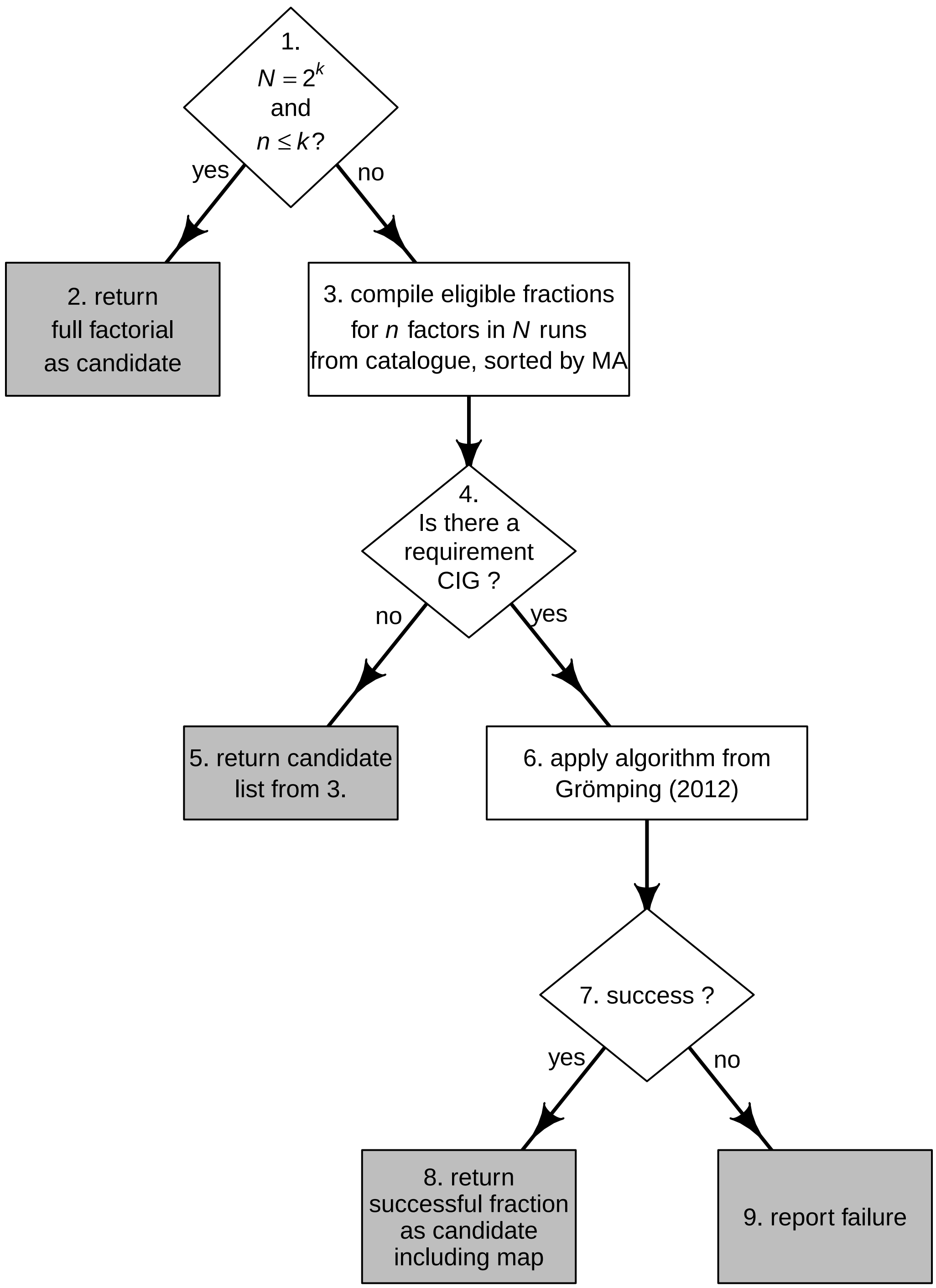

Figure 4: The algorithm for selecting one or more candidate fraction(s), given n, N, q, and possibly a requirement CIG

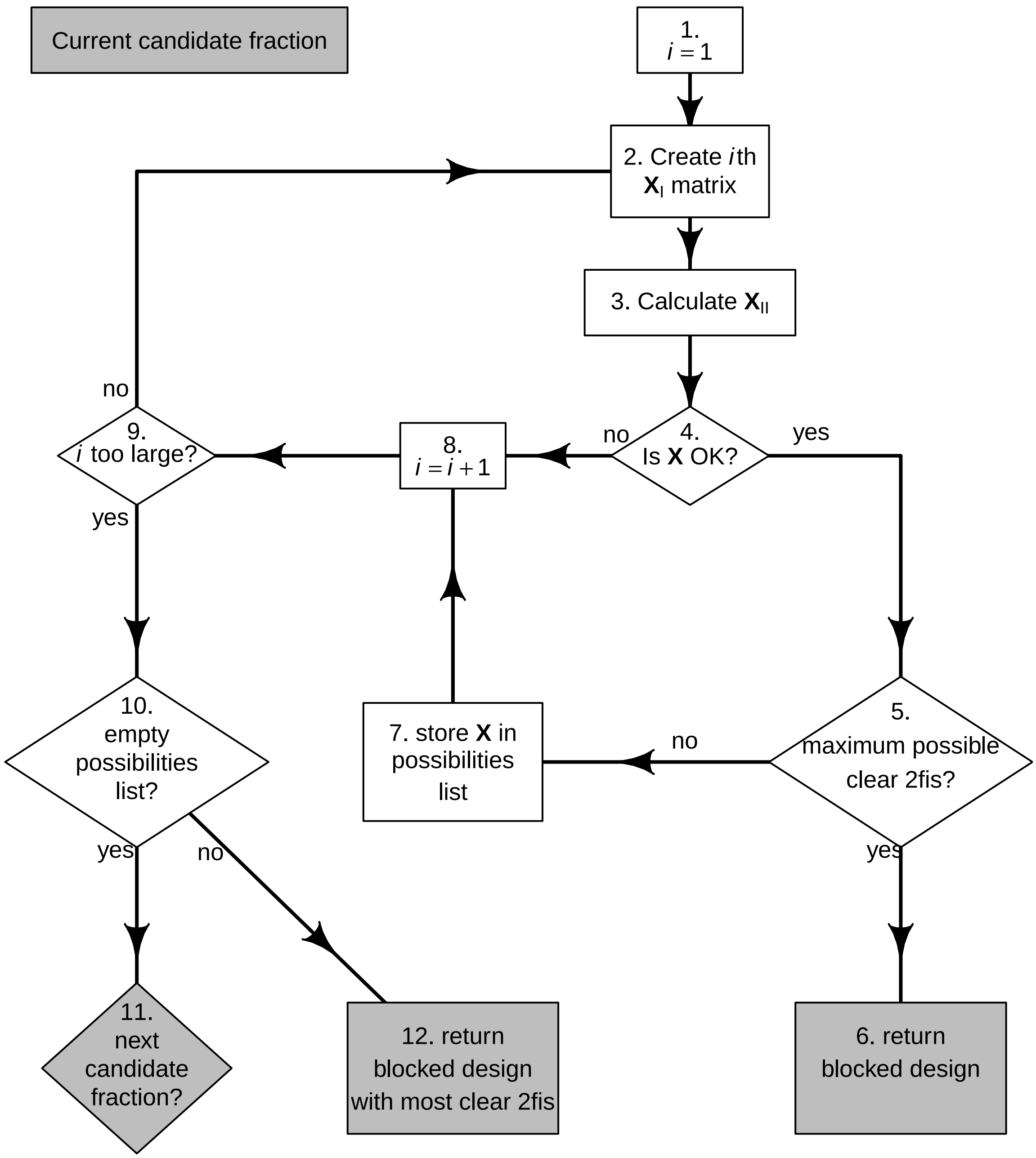

Figure 5: The algorithm for blocking a candidate fraction. For keeping the diagram simple, the check in step 4 includes the check for the requirement CIG, if applicable. This check involves a subgraph isomorphism check whether the requirement CIG is a subgraph of the estimability CIG of the blocked design.

$n = 200$ factors (which would imply a full factorial in 1.606938×10^{60} runs), function `colpick` constructs a suitable $\mathbf{X}$ matrix in about 1.5 seconds (for $q = 1$; larger q are faster, although $\mathbf{X} = \mathbf{1}_{200}^\top$ is trivial for $q = 1$). Creating the resulting blocked design from a given $\mathbf{X}$ breaks down for much smaller n: Blocking into blocks of size 2 ($q = 1$) works for up to $n = 16$ factors (i.e. 65536 runs), with run times exponentially increasing with n from under one second for up to 12 factors over 1.2s, 3.1s, 9.4s to 33.5s user time for 16 factors (for 16 factors, elapsed time is almost twice the user time; presumably a lot of time is lost for handling RAM limitations). These timings are from using the function `colpick` for creating an $\mathbf{X}$, followed by the function `FF_from_X` for creating the blocked design. For $q = 2$ or $q = 3$, times were somewhat faster, and the $n = 17$ factor case was still feasible (but $n = 18$ failed). The function `FrF2` works for up to 12 factors only and was much slower than the combination of `colpick` and `FF_from_X`, due to a larger overhead, especially for the large cases.

4.2 Full factorial with a requirement CIG

The automatic algorithm starts from the default without a requirement CIG, and afterwards loops through $\mathbf{X}$ column choices according to the approach described in Section 3.2, with the first components moving fastest. For each $\mathbf{X}$, a subgraph isomorphism check will indicate whether the requirement CIG can be accommodated in the estimability CIG. In the case of success, the algorithm returns a blocked design with the maximum possible number of clear 2fis that arises from the most balanced partition of the requirement CIG (see e.g. Example 4).

The algorithm fails if the requirement CIG has a chromatic number larger than $2^q - 1$ (e.g. with any non-empty requirement CIG for blocks of size 2, or if the requirement CIG contains a clique of size larger than $2^q - 1$), or for resource reasons from design creation (like in Section 4.1). Blocking may be slow if the requirement CIG is such that early $\mathbf{X}$ choices cannot accommodate the blocking. As an indication of worst case time consumption, timings have been taken for an impossible requirement CIG (a clique of size 2^q). For $q = 1$, because there is only a single possible $\mathbf{X}$ matrix (a row of ones) and the requirement CIG does not have any edges, infeasibility is confirmed fast regardless of n. For $q = 2$, the time needed for confirming infeasibility of the blocking is much longer and depends on n, because both the number of graphs and the size of each graph depend on n. Timings, obtained using the LAD algorithm of the package **igraph** (`method="LAD"` argument), again increase exponentially with n: the number of graphs triples with each additional factor, and the run time also roughly triples correspondingly; run times are shown in Table 3 in Section 4.4 as the special case with $p = 0$ ($k = n$). They were, for example, about 0.9 seconds for $n = 7$, 8.6 seconds for $n = 9$ and 91.4 seconds for $n = 11$, and about 288 seconds for $n = 12$ (larger n are possible but more time consuming, and have been omitted because they cannot be fractionated due to limitations of the function `FrF2`). Contrary to the case of a full factorial without a requirement CIG, here the time is needed for (not) finding the $\mathbf{X}$ matrix, instead of for constructing the blocked design from a given $\mathbf{X}$ matrix. Note that the full factorial is a very difficult case for the algorithm, because (a) all potential $\mathbf{X}$ matrices are feasible and (b) the estimability CIG for each $\mathbf{X}$ matrix has a large number of edges and is relatively regular (see resource considerations in Grömping 2012). At the same time, it may be possible for the user to create a suitable $\mathbf{X}$ matrix from inspecting the structure of the requirement CIG, because there is no confounding structure to be considered. Where this is possible, users may want to manually create a suitable $\mathbf{X}$ matrix with desired properties, e.g. using the R function `X_from_parts`, and to create the blocked design using the function `FF_from_X`.

4.3 Fractional factorial without a requirement CIG

For $k = n - p$, the sequence of $q \times k$ $\mathbf{X}_\mathrm{I}$ matrices is obtained in the same way as the sequence of $q \times n$ $\mathbf{X}$ matrices for the full factorial case. $\mathbf{X}_\mathrm{I}$ matrices for which $\mathbf{X}_\mathrm{II} = \mathbf{X}_\mathrm{I}\mathbf{Z}^\top$ contains any all-zero columns are discarded, as are $\mathbf{X}_\mathrm{I}$ matrices for which $\mathbf{X} = (\mathbf{X}_\mathrm{I} : \mathbf{X}_\mathrm{II})$ has fewer than q different columns. If all $\mathbf{X}_\mathrm{I}$ matrices are exhausted without finding a suitable blocking, the next candidate fraction is processed. In the case of success, the algorithm returns the blocking with the largest number of clear 2fis from the first (=MA) candidate for which a blocking was found. There may be fractions with worse aberration that would allow more clear 2fis. The code for Example 3 in Section C of the supplementary material exemplifies how to check whether the number of clear 2fis can be improved.

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

Blocking of fractions can at present only be handled with function `FrF2` and is thus limited to $k = n - p \leq 12$, i.e. up to 4096 runs. Within this limit, computing resources are not problematic (the largest case takes between three and four seconds on my machine). The algorithm will not fail, as long as $n \leq 2^{k-1}$ (i.e., a resolution IV fraction exists) and $1 \leq q \leq k - 1$. For too large p or too small q, there may be no or too few clear 2fis.

Availability of catalogued fractions can be an issue: the catalogue `catlg` within the R package **FrF2** is complete for up to 64 runs (for up to 32 runs from resolution III, for 64 runs from resolution IV), but does not contain all resolution IV fractions for 128 or more runs. Therefore, while blocking into blocks of sizes $2, \ldots, 2^{k-1}$ works and is guaranteed to be based on the MA fraction that enables the blocking for up to 64 runs, things are a bit more complicated for 128 or more runs:

- For up to 11 factors, the situation is like for 64 runs: all resolution IV or higher fractions are in the catalogue.
- For 12 to 24 factors, the catalogue `catlg` contains *a selection of* dominating fractions only (see Grömping 2014b). In particular, fractions without odd words are always dominated and were therefore deliberately omitted from the catalogue in 2013. However, as was observed in Section 3.2, keeping main effects clear from confounding with the block effect requires absence of odd-length words in applications with blocks of size 2. Therefore, the fractions that have MA among those without odd words have now been included into the catalogue `catlg`. These fractions allow blocking into blocks of size 2, and hence also blocking into larger block sizes. It is thus guaranteed that the desired block size is supported by the catalogue `catlg`. However, since the largest number of clear 2fis may arise from a non-catalogued fraction, the built-in catalogue `catlg` may not yield the best possible solution (see e.g. Example 7).
- The auxiliary package **FrF2.catlg128** contains a much larger set of fractions for 12 to 25 factors: it contains all even/odd fractions including the dominated ones, but omits the fractions without odd words (even the MA ones that were added to `catlg`). This catalogue is more likely to provide the best possible blockings for block sizes larger than 2, but it has not been systematically investigated whether this is always the case.
- A prior version of the auxiliary package **FrF2.catlg128** that contains *all* fractions for up to 24 factors is available from the author's homepage; they can be loaded as a package, or individual catalogues can be loaded, in order to make sure that no opportunity for a better blocking is missed. There are more fractions than in the current package, particularly for many factors (a few examples: 249 fractions instead of 179 for 12 factors, 3522 fractions instead of 2926 for 15 factors, 25064 fractions instead of 19466 for 18 factors, 82496 fractions instead of 39201 for 21 factors or 256654 fractions instead of 28133 for 24 factors). So far, no example for blocks of size 4 or larger was found for which use of a dominated fraction without words of length 5, i.e. use of the old catalogue, improved the result. However, a systematic investigation has not been conducted. Run time wise, checking all fractions from these complete catalogues will be prohibitive for larger cases; for the 13 factor case of Example 3, checking the 623 fractions (instead of 486 in the current package **FrF2.catlg128**) took about 25 minutes.
- For more than 128 runs or 128 runs with more than 25 factors, the catalogue `catlg` is currently the only available source for **FrF2**; it provides the overall MA fractions only. This limits the possibilities regarding the number (and structure) of clear 2fis (see e.g. Example 2).

If a situation is encountered for which the catalogue does not provide a suitable candidate for blocking (no blocking possible or no/too few clear 2fis for the blocked design), users can take one of two different routes:

- If an external catalogue is available (currently only in the package **FrF2.catlg128**), users can search through its fractions for finding a suitable blocking. For example, blocking the only fraction in 128 runs for 25 factors of catalogue `catlg` (fraction 25–18.1) into blocks of size 4 is not possible; one can search through the fractions from the larger catalogue `catlg128.25` from the package **FrF2.catlg128** (which holds 20569 fractions); searching through the first 1000 of these yielded at most 36 clear 2fis for the 988th element of the catalogue.
- Users who shy away from working with the external catalogue might also attempt to block a catalogued fraction for more factors, which may sometimes be successful; for example, while fraction 25–18.1 cannot be blocked into blocks of size 4, the larger fraction fraction 26–19.1 can, however, with only much fewer clear 2fis, since the fraction itself doesn't have any. One may be able to ascertain a few clear 2fis by

Accepted Manuscript for https://doi.org/10.1016/j.csda.2020.107059 (CSDA 2021 Vol. 153 pp. 1-18)

omitting a well-chosen column when accommodating the 25 treatment factors; however, if clear 2fis are important, the approach from the previous bullet is clearly preferable.

- If no external catalogue is available but suitable generators can be found outside of the catalogue, these can be used directly in function `FrF2`, or function `makecatlg` can be used for creating a catalogue from them, in order to provide functions `colpick` and/or `FrF2` with the required information for deriving a blocking, if possible. An example can be found in the supplementary material (Example 8 in Section D).

4.4 Fractional factorial with a requirement CIG

Initially, the internal function `mapcalc` applies the algorithm of Grömping (2012) for finding a fraction that can accommodate the requirement CIG (step 6 of Figure 4). Subsequently, the algorithm loops through $\mathbf{X}_\mathrm{I}$ matrices in the same way as in Section 4.3, with an additional subgraph isomorphism check to see whether the blocked design obtained with the $\mathbf{X}$ matrix can accommodate the requirement CIG. For resolution IV fractions, a separate subgraph isomorphism check for each $\mathbf{X}$ matrix can be switched off; in that case, the algorithm attempts to block the current fraction in the map order obtained from the initial check in step 6 of the algorithm of Figure 4. While this may save computational effort and may thus make some cases feasible that would otherwise lead to resource problems, it does sacrifice potential and will therefore not be further discussed. If the algorithm is successful, the blocked design is based on the MA element of the eligible fractions provided in step 3 of the algorithm of Figure 4 whose unblocked version is able to accommodate the requirement CIG. The number of clear 2fis in the successfully blocked design is maximal among all possible $\mathbf{X}$ matrices for blocking this fraction. There may be another fraction with worse aberration which can accommodate the requirement CIG with the requested block size while yielding more clear 2fis (see Example 7).

According to Lemma 3.2, failure of the algorithm can mean at least one of three things:

(i) There is not even an unblocked candidate fraction (i.e., the algorithm of Figure 4 has failed to return a candidate).

(ii) Blocks of size 2^q have been requested, but the requirement CIG has a chromatic number higher than $2^q - 1$.

(iii) Even though none of the above hold, a suitable $\mathbf{X}$ matrix without all-zero columns cannot be found based on the current candidate fraction that was returned by the algorithm of Figure 4.

Failures of type (i) produce an error message that starts with `The required interactions cannot be accommodated clear of aliasing ...`. Such failures can only be solved by increasing the number of runs or removing some edges from the requirement CIG, or by enhancing a catalogue or providing a custom catalogue using the function `makecatlg` (see Example 8 in the supplementary material, which provides a non-MA fraction from the literature for 256 runs, for which the built-in catalogue only contains MA fractions n–p.1). Algorithmic resources for this type of failure were discussed in Grömping (2012).

Failures of types (ii) or (iii) produce an error message that starts with `no adequate block design found`. Whether `no adequate block design` was found because of a failure of type (ii) or (iii) can be visually checked by plotting the requirement CIG (e.g. with the function `CIG` from the package **FrF2**, ideally in interactive mode). If the requirement CIG is $(2^q - 1)$-colourable, the failure must be of type (iii). There is currently no reliable algorithm in R that automates this check. In terms of resources, these failures can often be detected faster than in the full factorial situation, because (a) there are fewer candidate $\mathbf{X}$ matrices than in the full factorial situation ($k < n$), (b) part of the potential $\mathbf{X}$ matrices are ruled out because of the confounding structure and (c) the estimability CIGs may have fewer edges for resolution IV candidate fractions. Table 3 shows timings for impossible blocking requests of full factorials and half to 256th fractions with run sizes 2^k ($k = 5$ to $k = 12$). As the preceding considerations imply, the time used to identify impossibility of the request decreases with increasing fractionation.

A failure of type (iii) implies that the blocking task might be solvable with a different candidate fraction. For attempting a solution along this road, the algorithm of Figure 4 can be started again after removing the failed fraction from the eligible fractions in step 3. We already met an example of this kind: in Example 7 with 13 factors to be studied in 128 runs in blocks of size 4, errors of type (iii) occurred for fractions 13–6.1 and 13–6.2, and were remedied by removing those fractions from the candidate list and repeating the algorithm

Table 3: Times[s] from the function `FrF2` for attempting to block a suitable $(k+p)$–p fraction into blocks of size 4, while keeping a clique of size 4 clear

k	run size	$p=0$	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$	$p=8$
5	32	0.12	0.12	0.13	0.02					
6	64	0.32	0.28	0.23	0.23	0.30	0.23			
7	128	0.91	0.75	0.64	0.53	0.47	0.58	0.66	0.41	0.43
8	256	2.67	2.46	1.97	1.63	1.39	1.15	0.99	0.90	0.82
9	512	8.64	7.16	5.89	5.19	4.06	3.56	3.24	3.05	2.91
10	1024	28.17	23.89	18.81	15.27	12.50	10.34	8.75	7.70	6.84
11	2048	91.36	72.41	59.92	48.57	40.22	33.42	27.35	22.61	18.86
12	4096	287.98	252.67	206.85	165.66	131.00	110.97	92.20	78.42	68.66

(see code in the supplementary material). Where such a restart is necessary, it is useful to include dominated fractions (e.g. 13–6.3) into the list of eligible candidates in the algorithm of Figure 4.

It is somewhat inconvenient that the automatic search is restricted to only a single fraction that can accommodate the requirement CIG and fails if this fraction cannot be successfully blocked. After gaining more experience with the algorithm, this decision may be revised. For the time being, automation is possible outside of the function `FrF2`: Section C of the supplementary material shows code for Example 7 with which a search over more fractions can be automated. That code also exemplifies how to use catalogues from R package **FrF2.catlg128**, which is necessary because the most successful fraction 13–6.3 is not in catalogue `catlg`, but only in the larger catalogue `catlg128.8to15` from that package.

Availability of catalogues is an even more critical issue than for the creation of designs without a requirement CIG. This topic has been discussed at length in the previous subsection. Whenever an external set of generators is available, function `makecatlg` allows users to inspect which 2fis are kept clear by the set; if this looks promising, a blocking attempt can be made.

5 Discussion

Godolphin (2019) proposed a powerful tool that allows blocking to be combined with keeping a set of selected 2fis (the requirement CIG) clear, or with maximizing the number of clear 2fis. She provided a selection of templates for obtaining blocks of size 4 for selected situations with up to 128 runs and up to 12 factors. If one of the templates suits their needs, practitioners can use these for accommodating their combined estimability and blocking requirements. This paper presents an algorithm for combining estimability and blocking needs in more general situations. If a requirement CIG is specified, the proposed algorithm uses the algorithm by Grömping (2012) for selecting a candidate fraction from a sorted catalogue of fractions, which also holds each fraction's estimability CIG. The subsequent application of the blocking approach by Godolphin makes use of subgraph isomorphism checking again and returns the most adequate blocking of this fraction, if one exists. Otherwise, the process has to be repeated with the next suitable fraction. The algorithm is implemented in the R package **FrF2**. It is fully supported by the catalogue in that package for all settings with up to 64 runs and for 128 runs with up to 11 factors. For 128 runs with 12 to 25 factors, auxiliary catalogues are available. For larger designs, published generators from the literature (e.g. from Xu 2009) can be incorporated (see Example 8 in Section D of the supplementary material).

For unblocked fractions, MA is universally accepted as a suitable quality criterion. Quality criteria have also been proposed for blocked designs (see Godolphin 2019 an references therein). However, there is no criterion that is as universally agreed as the MA criterion for unblocked designs. For example, Sun, Wu and Chen (1997) discussed that a suitable design must be adapted to the experimental situation. The blocked designs returned by the proposed algorithm have not been investigated in terms of any adapted aberration criteria for blocked designs. Rather, they are based on the MA unblocked fraction for the treatment factors, such that the requirements (block size and possibly a requirement CIG) can be accommodated; within the successful unblocked fraction, blocking is conducted such that treatment 2fis are kept clear as much as possible.

Where a requirement CIG is to be kept clear, it is inconvenient that the automatic algorithm searches only a single suitable candidate and currently has to be manually (or programmatically) restarted if this candidate proves unsuitable for the desired blocking, or for comparing all possible solutions in terms of further criteria. A future version of the algorithm may provide an automation. Another important improvement concerns the availability of catalogues for larger designs: for 256 runs and more, a lack of catalogued fractions limits the applicability of the algorithm to find the best possible blocking for the situation at hand, as there are often better choices than the MA fraction. Catalogues of more fractions in 256 runs – e.g. the ones available from Hongquan Xu's website in support of his 2009 paper – may be made available for R package **FrF2** (possibly in another separate auxiliary package), for supporting the blocking methodology. If, in practice, it is infeasible to use very large catalogues, then their availability is an asset in theory only. This matter will be further investigated.

The entire paper focused on constructing a block factor such that treatment factor main effects are unconfounded with the block factor main effect. However, with a large number of small blocks in a relatively large design, one can also adopt an approach that allocates some factors at the block level only. While this is detrimental for the inference regarding the affected factors, it may still be a suitable route to choose and may sometimes even be unavoidable (e.g., because some factors can only be changed at the block level). Designs of this nature are called split-plot designs. It is of interest to investigate whether their automated creation can also benefit from using Godolphin's approach based on $\mathbf{X}$ matrices with selected all-zero columns.

References

Block, R. and Mee, R. (2005). Resolution IV designs with 128 runs. *Journal of Quality Technology* **37**, 282-293.

Block, R. and Mee, R. (2006). Corrigenda. *Journal of Quality Technology* **38**, 196.

Bondy, J.A. and Murty, U.S.R. (1982). *Graph Theory with Applications.* North-Holland, New York.

Chen, B., Li, P., Liu, M.-Q. and Zhang, R. (2006). Some results on blocked regular 2-level fractional factorial designs with clear effects. *Journal of Statistical Planning and Inference* **136**, 4436-4449.

Chen, J., Sun, D.X. and Wu, C.F.J. (1993). A catalogue of 2-level and 3-level orthogonal arrays. *Int. Statistical Review* **61**, 131-145.

Cordella, L.P., Foggia, P., Sansone, C. and Vento, M. (2001). An improved algorithm for matching large graphs. *Proc. of the 3rd IAPR TC-15 Workshop on Graphbased Representations in Pattern Recognition*, 149-159.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems* 1695.

Godolphin, J. (2019). Construction of blocked factorial designs to estimate main effects and selected two-factor interactions. *Unpublished report.* https://arxiv.org/abs/1907.02373

Grömping, U. (2012). Creating clear designs: a graph-based algorithm and a catalog of clear compromise plans. *IIE Transactions* **44**, 988-1001. DOI:10.1080/0740817X.2012.654848.

Grömping, U. (2014a). R package FrF2 for creating and analysing fractional factorial 2-level designs. *Journal of Statistical Software* **56** (1), 1-56.

Grömping, U. (2014b). A note on dominating fractional factorial two-level designs with clear two-factor interactions. *Technometrics* **56** (1), 42-45.

Mee, R. (2009). *A Comprehensive Guide to Factorial Two-Level Experimentation*. Springer-Verlag, New York.

Ryan, K.J. and Bulutoglu, D.A. (2010). Minimum aberration fractional factorial designs with large N. *Technometrics* **52**, 250-255.

Solnon, C. (2010). AllDifferent-based filtering for subgraph isomorphism. *Artificial Intelligence* **174**, 850-864.

Sun, D.X., Wu, C.F.J. and Chen, Y.Y. (1997). Optimal blocking schemes for 2^p and 2^{n-p} designs. *Technometrics* **39**, 298-307.

Wu, H., Mee, R., and Tang, B. (2012). Fractional factorial designs with admissible sets of clear two-factor interactions. *Technometrics* **54** (2), 191–197.

Xu, H. (2009). Algorithmic construction of efficient fractional factorial designs with large run sizes. *Technometrics* **51**, 262-277. Online supplement at http://www.stat.ucla.edu/%7Ehqxu/pub/ffd2r/.

Zhao, S., Li, P. and Liu, M.-Q. (2013). On blocked resolution IV designs containing clear two-factor interactions. *Journal of Complexity* **29**, 389-395.