Fachbereich II – Mathematik - Physik - Chemie

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN

University of Applied Sciences

03/2019

Ulrike Grömping

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

Ein Algorithmus zur Blockbildung in regulären fraktionierten faktoriellen Plänen mit 2-stufigen Faktoren mit unvermengten Zweifach-Wechselwirkungen

Reports in Mathematics, Physics and Chemistry

Berichte aus der Mathematik, Physik und Chemie

ISSN (print): 2190-3913

Reports in Mathematics, Physics and Chemistry

Berichte aus der Mathematik, Physik und Chemie

The reports are freely available via the Internet:
http://www1.beuth-hochschule.de/FB_II/reports/welcome.htm

03/2019, August 2019

An algorithm for blocking regular fractional factorial 2-level designs with clear two-factor interactions
Ein Algorithmus zur Blockbildung in regulären fraktionierten faktoriellen Plänen mit 2-stufigen Faktoren mit unvermengten Zweifach-Wechselwirkungen (englischsprachig)

Update history
August 18 2019: An earlier version of August 16th has been replaced because of a strategy improvement in the algorithm (subgraph isomorphism checks for every X). The previous version was archived by its author and is available from her on request.

Editorial notice / Impressum

Published by / Herausgeber:
Fachbereich II
Beuth Hochschule für Technik Berlin
Luxemburger Str. 10
D-13353 Berlin
Internet: http://public.beuth-hochschule.de/FB_II/
E-Mail: fbiireports@beuth-hochschule.de

Responsibility for the content rests with the author(s) of the reports.
Die inhaltliche Verantwortung liegt bei den Autor/inn/en der Berichte.

ISSN (print): 2190-3913

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

Ulrike Grömping

Beuth University of Applied Sciences, Berlin

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. This paper provides an algorithm 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 combines the algorithm of Grömping (2012) with an implementation of a recent proposal by Godolphin (2019) for the blocking. It is implemented in the R package **FrF2**.

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. A recent article by Janet Godolphin (2019) has opened an interesting 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. This paper provides an algorithm for the automation of the procedure, 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). It incorporates 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. The justification of Godolphin's blocking method is closely related to graph-colouring and multipartite graphs; its algorithmic implementation does not make use of graph methods related to graph colouring, but does make use of subgraph isomorphism checking in case specific 2fis are requested to be clear.

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 deep-dives the approach by Godolphin for blocking full factorial designs, without or with requesting specific 2fis to be clear, and Section 4 presents the extension to blocking fractional factorial designs of resolution at least IV. Section 5 describes the algorithm and its implementation in package **FrF2**. The final discussion points out limitations and opportunities for future extensions. An appendix provides details on the fractions used in this paper, an overview of relevant R functions in package **FrF2**, R code for the examples from Sections 3, 4 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

Table 1: The confounding pattern from defining contrasts E=ABC and F=ABD. Headers give base 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

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 can be obtained by specifying $p > 0$ defining contrasts, which declare how p factors can be added to a full factorial in $k = n - p$ 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. Defining contrasts imply defining words (here: ABCE, ABDF), which are factor combinations whose sum is a constant $N \times 1$ vector with the neutral element 0, denoted as I; the p defining words span a word group of size 2^p (including I) 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: for each effect in the header, the table shows the confounded effects that can be deduced from the two defining words ABCE and ABDF and from the resulting word group element CDEF.

An important quality criterion for a fraction is the word length pattern (WLP), which is the frequency table of word lengths that occur in the word group (three words of length 4 in Table 1). The length of the shortest 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.

Each fraction has a confounding structure; isomorphic fractions have confounding structures that 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 R packages **FrF2** and **FrF2.catlg128**. Catalogued fractions are denoted as *n*-*p*.rank, where rank is the rank number in terms of the MA criterion (arbitrary in case of ties), i.e. rank 1 denotes the best fraction in terms of the MA criterion. Thus, the 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; uses m instead of n for the number of factors, g instead of p for the number of generators, and the multiplicative group {-1, +1} instead of the additive group {0, 1} modulo 2).

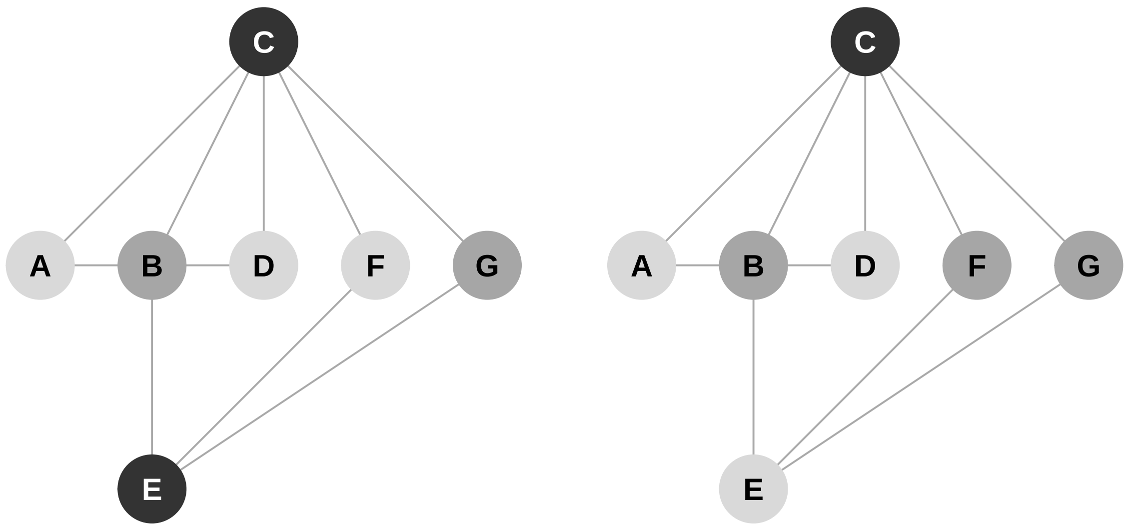

Figure 1: A requirement CIG with two different colourings.

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). 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. 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). 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 is an object that consists of $n > 0$ vertices some or all of which are connected by edges (it is conceivable but usually boring that a particular graph has no edges whatsoever). 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 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).

- 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 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.
- *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, $r < c \le n$.
- 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.
- 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 $V_1, \dots, V_r$ gives rise to a length r *r-profile* $P =< n_1, \dots, n_r >$, with $n_1 \ge \dots \ge n_r > 0$ 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.
- 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 for the 3-partite graph of Figure 1, the two colourings of which correspond to partitions {{A,D,F}, {B,G}, {C,E}} or {{A, D, E}, {B, F, G}, {C}}. The 3-profiles of these 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 3 and 7, where Figure 1 will be re-visited).

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^{n-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 $N/2$ ($q = n - 1$) or four blocks of size $N/4$ ($q = n - 2$). However, it can happen that only smaller blocks can be made homogeneous enough, for example in 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 assumed to be active so that treatment effects confounded with a block degree of freedom cannot be estimated.

In line with Godolphin (2019), this paper only looks at 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, because interactions between block factors are assumed to be as important as block main effects. Thus, a situation with four days and two shifts per day would be handled as a single block factor with eight blocks.

There are two principal approaches for blocking an existing design:

- specify p_{block} block generators in order to get a block factor with $2^{p_{\text{block}}}$ levels,
- specify a principal block of size 2^q as a sub group of the N runs and obtain 2^{n-p-q} blocks of size 2^q as this group and all its cosets.

R package **FrF2** takes the first approach, Godolphin (2019) used the second approach. The two approaches are equivalent and can be used for constructing exactly the same block structures. Thinking about construction in terms of Godolphin's approach is useful especially for small block sizes, because these correspond to small q.

2.5 Basics of R package FrF2

All calculations for this paper have been done with Version 2.1 of R package **FrF2**; an earlier version was described in Grömping (2014a). Function `FrF2` produces regular fractional factorial 2-level designs. Its most basic form is to specify `nruns=` $N = 2^{n-p}$ and `nfactors=` n, which will yield an experimental design based on the MA fraction n–p.1. This is possible, because the package contains a large catalogue (`catlg`) of nonisomorphic fractions, which are sorted from good to bad in terms of the MA criterion.

Function `FrF2` allows 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 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 package **igraph** (Csardi and Nepusz 2006).

Function `FrF2` furthermore allows 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 Godolphin's approach for blocking full factorial designs

This section discusses blocking an unreplicated full factorial design in n factors into 2^{n-q} blocks of size 2^q. As preliminary notation, $\mathcal{X}_q$ 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 $2^q - 1$.

Godolphin's idea for block construction is simple: compile a $q \times n$ matrix $\mathbf{X}$ such that each column is an element of $\mathcal{X}_q$. Construct a "principal block" of size 2^q from $\mathbf{X}$, and obtain the other blocks as cosets of this principal block.

Example 1: For blocking a full factorial in 7 factors (and thus 128 runs) into 16 blocks of size 8 ($n = 7$, $q = 3$),

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

can be used. The rows of $\mathbf{X}$ generate all eight level combinations for the principal block: these are

- an all-zero row (denoted as (1) in Godolphin 2019, here: (0 0 0 0 0 0 0)),
- the rows of $\mathbf{X}$ themselves,

- and all sums from two or more rows of $\mathbf{X}$ (modulo 2) (here: four sum rows).

When denoting the factors by capital letters (here: A to G), the experimental runs are often denoted by corresponding small letters (here: a to g) for those factors that take on the value 1 in the run. In this notation, the principal block thus consists of rows (1), adefg, bceg, bcfg, abcdf, abcde, ef, adg. The other 15 blocks can then be constructed as cosets of this principal block, by adding 15 independent constant runs to the principal block. Since this paper takes a different approach for constructing the blocks from the matrix $\mathbf{X}$, dealing with cosets is not further considered.

For finding all treatment factor effects that are confounded with the block factors, denote as $\mathbf{F}$ the $2^n \times n$ matrix of full factorial level combinations in 0/1 coding and consider each of its rows as the coding of an effect. When using the matrix $\mathbf{X}$ for blocking a full factorial design, the effects confounded with the block factor main effect are those that yield a column of zeroes in the matrix $\mathbf{X}\mathbf{F}^\top$.

Example 1, continued: The block factor must have $2^{7-3} = 16$ levels, i.e. 15 effects (in addition to the constant) must be confounded with its main effect. These are the factorial effects BC, AD, ABCD, BEF, CEF, ABDEF, ACDEF, ABG, ACG, BDG, CDG, AEFG, ABCEFG, DEFG, BCDEFG. We see that no main effect is confounded with blocks, but two 2fis are confounded with blocks.

From Godolphin (2019), we learn the rules behind this observation:

Lemma 3.1 (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 no all-zero columns, the following results hold:*

- *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.*

Example 1, continued: As the matrix $\mathbf{X}$ in (1) has full row rank and all its columns are non-zero, main effects are not aliased with blocks. Columns 1 and 4 of $\mathbf{X}$ are identical, as are columns 2 and 3; thus, AD and BC are confounded with the block factor. All other 2fis are unconfounded with blocks.

Example 2: In the situation of Example 1, Lemma 3.1 implies that we can improve the blocking by choosing seven different columns for $\mathbf{X}$ (which is possible here, because $n \le 2^q - 1 = 7$). With

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{2}$$

we get the preferable block confounded set BCD, ABCE, ADE, ABF, ACDF, CEF, BDEF, ACG, ABDG, BEG, CDEG, BCFG, DFG, AEFG, ABCDEFG and can thus keep all 2fis clear.

For actually creating the blocked design, $k - q$ independent block generators are needed: In Examples 1 and 2, as blocks of size $2^3 = 8$ are needed, the block factor must have $2^{7-3} = 16$ levels. It can be created using any four independent block confounded effects as generating contrasts, e.g. BCD, ABCE, ABF, ACG for Example 2.

Each column of the $\mathbf{X}$ matrix corresponds to an experimental factor. The $\mathbf{X}$ matrix thus partitions the 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. The sorted vector of part sizes is called the *profile* of $\mathbf{X}$. For example, the profiles induced by the $\mathbf{X}$ matrices (1) and (2) are <2,2,1,1,1,0,0> and <1,1,1,1,1,1,1>, respectively. Zeroes are often omitted from the profile.

3.1 Profiles and estimability CIGs

Lemma 3.1 implies that the estimability CIG of a full factorial blocked according to a rank q $\mathbf{X}$ matrix without all-zero columns is a complete r-partite graph, where r is the number of different columns in $\mathbf{X}$ (at most $2^q - 1$). Correspondingly, there is a unique partition of the estimability CIG into r independent sets, which coincides with the partition obtained from $\mathbf{X}$, and the r-profile of the blocked design's estimability CIG equals the profile of $\mathbf{X}$. A graph of the 2fis that are aliased with blocks contains exactly the independent sets from the estimability CIG as cliques. Figures 2 and 3 visualize both the aliased

2fis and the estimability CIGs for four different 3-profiles for 13 factors: Figure 2 shows the 2fis that are confounded with blocks, while Figure 3 shows the estimability CIGs of the blocked designs. The edges present in Figure 2 are exactly those edges that are missing in Figure 3, and the factor partition corresponding to the profile directly implies which edges are of which type. All four CIGs are complete 3-partite graphs and thus have unique profiles. The colouring in Figure 2 corresponds to the colouring in Figure 3 and is (of course) not a proper colouring in the sense defined in Section 2.3.

3.2 Number of two-factor interactions unconfounded with the block main effect

With seven factors, it was possible to keep all 2fis clear in a full factorial with blocks of size 8. Had we wanted to block eight instead of seven factors into blocks of size 8, it would no longer be possible to keep all 2fis clear, because there are only $2^3 - 1 = 7$ potential vectors for $\mathbf{X}$. Godolphin (2019, Theorem 2) provided an upper bound for the number of clear 2fis from blocking a full factorial in n factors into blocks of size 2^q:

$$\phi_{\max} = \binom{n}{2} - vw - (2^q - 1)\binom{v}{2}, \tag{3}$$

with $v = [n/(2^q - 1)]$ and $w = n - (2^q - 1)v$, where $[\cdot]$ denotes the floor function.

For $n = 8$ and $q = 3$, i.e. 8 factors in blocks of size 8, Formula (3) yields $\phi_{max} = 27$; thus, at least one of the 28 possible interactions is confounded with blocks. With the $\mathbf{X}$ matrix approach, it is straightforward to influence which interaction is sacrificed: assign identical columns in $\mathbf{X}$ to those two factors whose interaction is least important.

In generalization of formula (3), Godolphin (2019) provided a formula for the number of clear 2fis in a blocked full factorial based on the profile $< n_1, \dots, n_{2^q-1} >$ (see end of her section 2): there are

$$\sum_{i=1}^{2^q-2} \sum_{j=i+1}^{2^q-1} n_i n_j \tag{4}$$

clear 2fis. It is straightforward to see this, as this 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}. For example, the complete 3-partite graph with the profile <6,6,1> has $6 \cdot 6 + 6 \cdot 1 + 6 \cdot 1 = 48$ clear 2fis (see also Figure 3). Formula (4) works without change, if only $r < 2^q - 1$ elements of $\mathcal{X}_q$ occur in $\mathbf{X}$, by setting profile components $n_{r+1}, \dots, n_{2^q-1} = 0$. Formula (3) gives the number of edges in a $2^q - 1$-partite graph with n vertices and 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. Thus, if interest is in obtaining a blocked design with as many clear 2fis as possible, the largest and smallest frequencies of columns of $\mathcal{X}_q$ used in $\mathbf{X}$ should differ by at most 1 (including zeroes for absent elements of $\mathcal{X}_q$).

Table 2 shows the numbers of 2fis that can be kept clear in full factorial designs for $n = 3, \dots, 28$ in blocks of sizes 2^2, 2^3 or 2^4 for the overall most balanced profile (these are simply the $\phi_{\max}$ values) and for the most balanced profile attainable that keeps all 2fis of a single factor or of two specific factors clear (headers <...,1> and <...,1,1>, respectively). For $n \leq 15$ factors, blocks of size 16 trivially allow a unique $\mathcal{X}_q$ element for each factor, so that all 2fis can be kept clear. For $n \leq 7$, this is also possible with block size 8 (see Example 2), and for $n = 3$ even with block size 4. Note that profiles <5,4,4> and <6,6,1> in Figures 2 and 3 correspond to the table entries for thirteen factors in blocks of size 4 in columns "best" and "<...,1>", respectively.

3.3 Keeping specific two-factor interactions clear

The most fundamental result is the following:

Lemma 3.2 (Godolphin, Lemma 1)**.** *The 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.*

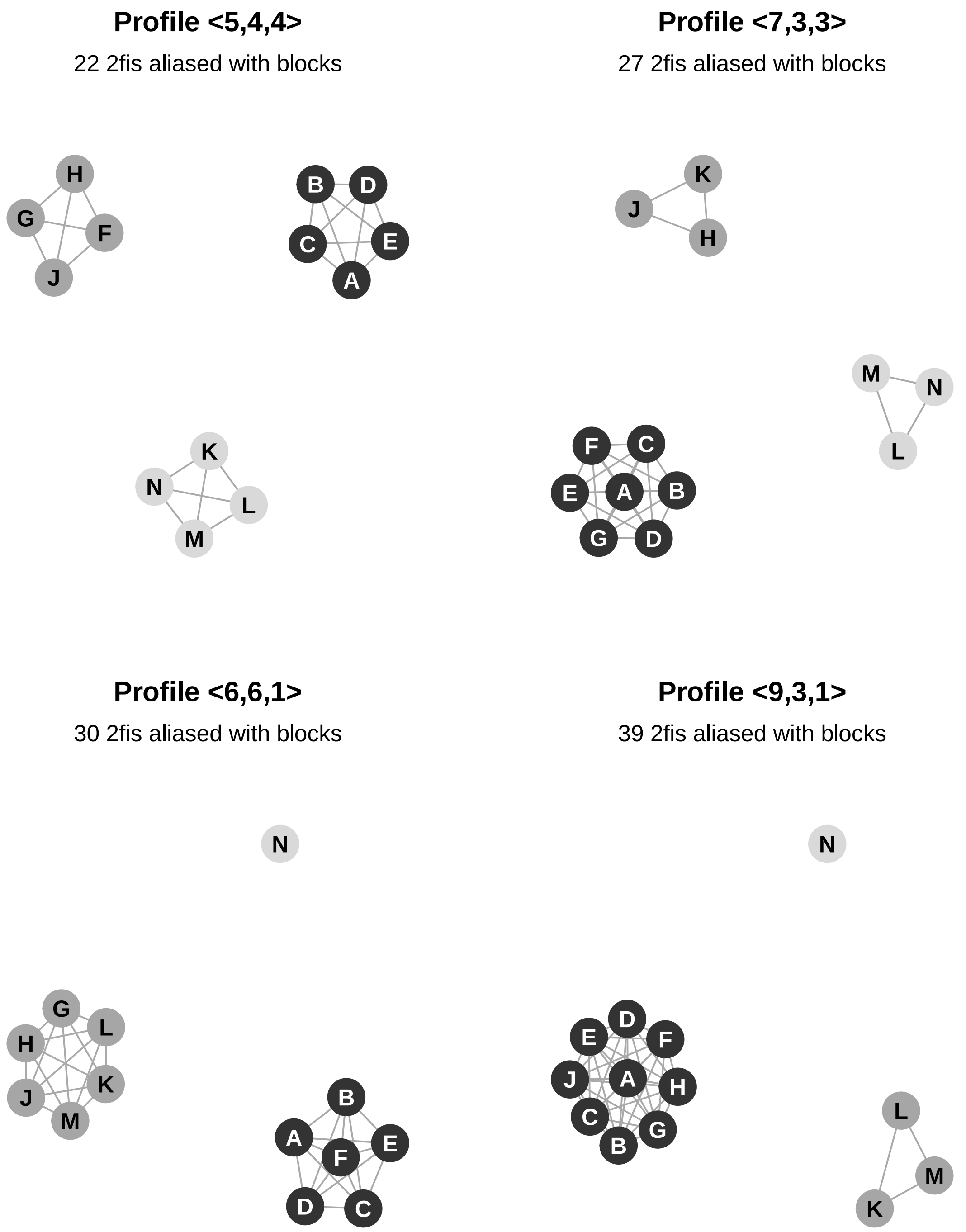

Figure 2: 2fis aliased with the block main effect for four different blockings of the 13 factor full factorial into blocks of size 4. Note: these are not CIGs, but the complements of CIGs.

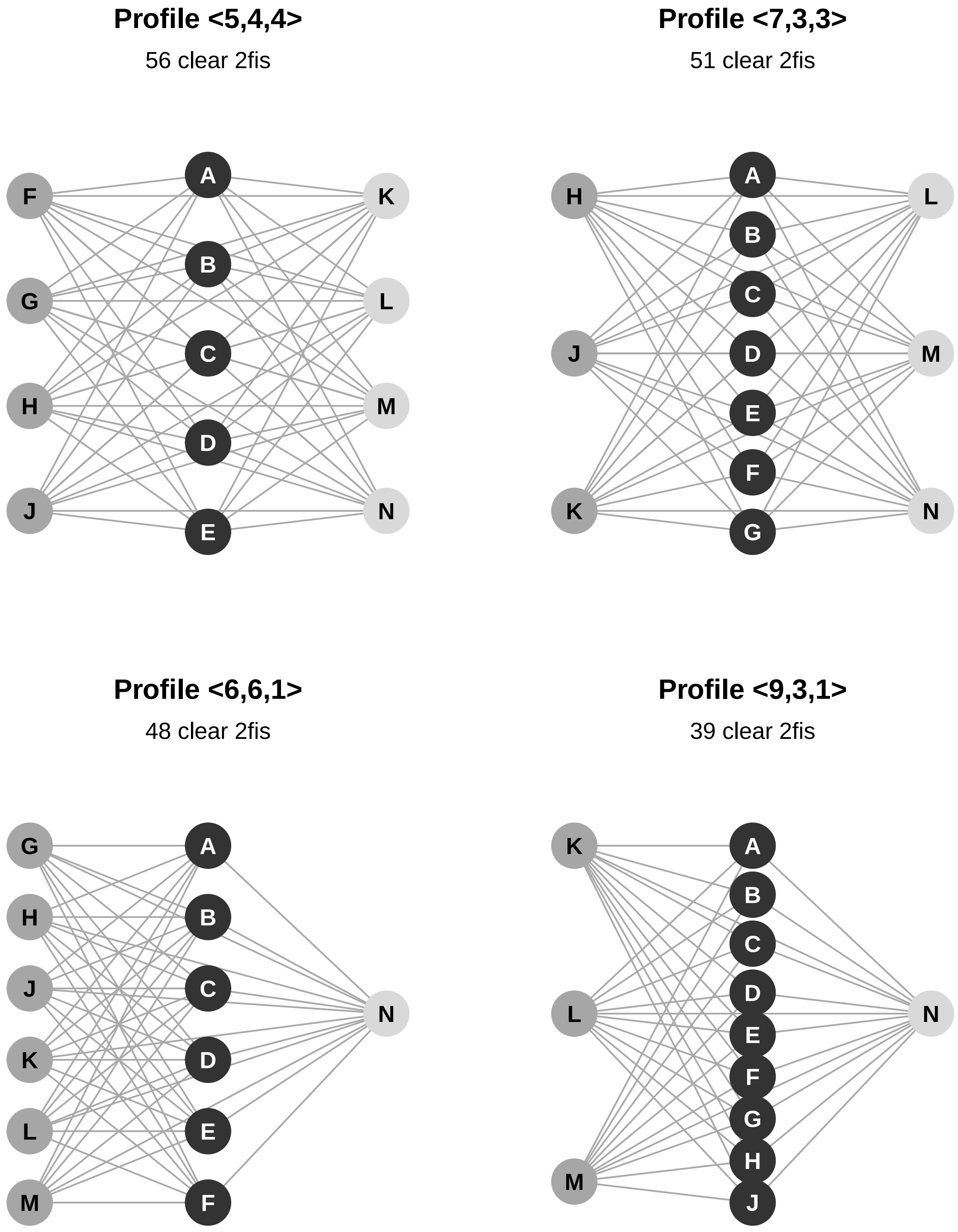

Figure 3: Estimability CIGs for the designs of Figure 2

Table 2: Maximum numbers of clear 2fis for n factors in blocks of sizes 4, 8 and 16 for three types of profiles

n	2fis	best		<...,1>		<...,1,1>		n	2fis	best			<...,1>			<...,1,1>		
		4	8	4	8	4	8			4	8	16	4	8	16	4	8	16
3	3	3	3	3	3	3	3	16	120	85	109	119	71	108	119	29	107	119
4	6	5	6	5	6	5	6	17	136	96	123	134	80	122	134	31	121	134
5	10	8	10	8	10	7	10	18	153	108	138	150	89	137	150	33	135	150
6	15	12	15	11	15	9	15	19	171	120	154	167	99	153	167	35	150	167
7	21	16	21	15	21	11	21	20	190	133	171	185	109	169	185	37	166	185
8	28	21	27	19	27	13	27	21	210	147	189	204	120	186	204	39	183	204
9	36	27	34	24	34	15	34	22	231	161	207	224	131	204	224	41	201	224
10	45	33	42	29	42	17	42	23	253	176	226	245	143	223	245	43	219	245
11	55	40	51	35	51	19	51	24	276	192	246	267	155	243	267	45	238	267
12	66	48	61	41	61	21	61	25	300	208	267	290	168	264	290	47	258	290
13	78	56	72	48	72	23	71	26	325	225	289	314	181	285	314	49	279	314
14	91	65	84	55	83	25	82	27	351	243	312	339	195	307	339	51	301	339
15	105	75	96	63	95	27	94	28	378	261	336	365	209	330	365	53	323	365

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.

If interest is in a particular set of clear 2fis, obtaining the largest possible number of clear 2fis may not be appropriate. For example, if one wants to conduct a full factorial in 13 factors in blocks of size 4 keeping all 2fis of two particular factors clear, the unique profile of the requirement CIG is <11,1,1>, and Table 2 tells us that there can only be 23 clear 2fis, which corresponds to exactly the 2fis with the two particular factors.

Hence, the construction of a suitable matrix $\mathbf{X}$ for accommodating the requirement CIG in the blocked design 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 3.1: if possible, it is always best to find a $2^q - 1$ partition of the requirement CIG (even if a smaller number of parts would suffice). Blocking the full factorial based on an $\mathbf{X}$ matrix that is compatible with this partition will then ensure that the requirement CIG can be accommodated in the resulting complete $2^q - 1$-partite estimability CIG of the blocked design. If the requirement CIG permits more than one $2^q - 1$-partition, the most balanced one should be chosen in order to maximize the number of clear 2fis in the blocked design.

Example 3: A full factorial design is to be blocked into blocks of size 4, such that the requirement CIG of Figure 1 is kept clear. The two colourings 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. The 3-partition corresponding to the profile is {{A,D,F},{B,G},{C,E}}. Using vectors $(0,1)^\top$, $(1,0)^\top$, and $(1,1)^\top$ in this order,

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

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.

There are some obvious restrictions to what is possible:

- Whenever a requirement CIG contains a clique of size c, its chromatic number is at least c. Thus, if for example all interactions within a set of four factors are requested to be clear, blocking into blocks of size $4 = 2^2$ is not possible, because the chromatic number of the requirement CIG exceeds $2^2 - 1 = 3$.
- If a requirement CIG contains all 2fis of a single factor, the chromatic number of the subCIG for the other $n - 1$ factors must be one less than the required chromaticity. This can be iterated; thus,

– if all 2fis involving two specific factors are to be estimated in blocks of size 4, the remaining factors must be singletons in the requirement CIG,
– if all 2fis involving three specific factors are to be estimated in blocks of size 8, the requirement subCIG of the remaining $n-3$ factors must be 4-colourable ($4 = 2^3 - 1 - 3$),
– and so forth.

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.

4 Blocking fractional factorial designs

This section deals with blocking a fraction of 2^{n-p} runs in n factors into 2^{n-p-q} blocks of size 2^q. The results from the previous section on full factorials remain valid, regarding which 2fis are confounded with the block factor, given a particular $\mathbf{X}$ matrix (see Lemma 4.1 below). It is, however, more complicated to obtain an $\mathbf{X}$ matrix that is suitable for creation of a principal block, and there may be confounding among 2fis even in the unblocked fraction. The following lemma states the consequences from blocking a fraction if a valid matrix $\mathbf{X}$ can be found:

Lemma 4.1. *Let $\mathcal{F}$ denote a $2^{n-p} \times n$ fraction for n factors, and let $\mathbf{Z}$ denote the $p \times n$ confounding matrix of $\mathcal{F}$. Suppose that a $q \times n$ matrix $\mathbf{X} = (\mathbf{X}_I | \mathbf{X}_I \mathbf{Z}^\top)$ with columns from $\mathcal{X}_q$ can be found. We have the following results:*

(i) $\mathcal{F}$ can be blocked into blocks of size 2^q without confounding main effects with blocks.
(ii) The estimability CIG of the blocked fraction is the intersection between the estimability CIG of $\mathcal{F}$ and the estimability CIG of the full factorial blocked according to $\mathbf{X}$.

4.1 Feasible X matrices and profiles, and numbers of clear 2fis

When blocking designs that already have confounding among their treatment factors, not every $\mathbf{X}$ matrix can be used for creating the principal block:

- Only $n-p$ columns of the $q \times n$ $\mathbf{X}$ matrix can be freely chosen from the columns of $\mathcal{X}_q$, w.l.o.g. the first ones. In line with Godolphin (2019), we denote the first $n-p$ columns of $\mathbf{X}$ as $\mathbf{X}_\mathrm{I}$.
- The remaining p columns of $\mathbf{X}$ – denoted as $\mathbf{X}_\mathrm{II}$ – must be calculated such that they comply with the design's confounding structure. This is done by applying the design's generating contrasts to the columns of the matrix $\mathbf{X}_\mathrm{I}$.

Example 4: 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. In analogy to Godolphin (2019), the generating contrasts are expressed by a $p \times (n-p) = 5 \times 8$ matrix $\mathbf{Z}$ in the obvious way:

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

From an arbitrary $q \times (n-p) = 2 \times 8$ matrix

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

(columns labelled with letters A to H), the corresponding $q \times p = 2 \times 5$ matrix

$$\mathbf{X}_\mathrm{II} = X_\mathrm{I} \mathbf{Z}^\top = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{8}$$

(columns labelled with letters J to N) is obtained, and the entire matrix is then $\mathbf{X} = (\mathbf{X}_\mathrm{I} \vdots \mathbf{X}_\mathrm{II})$. Remember that matrix multiplication is modulo 2. We see an all-zero column in $\mathbf{X}$ in the position of factor K. Thus, blocking a full factorial with this $\mathbf{X}$ matrix would confound the main effect of factor K with the block effect, which is undesirable (but see the discussion). Hence, $\mathbf{X}_\mathrm{I}$ from (7) is not suitable for blocking the fraction.

Unfortunately, it is by no means straightforward to specify sufficient conditions for the existence of a matrix $\mathbf{X}$ that is assumed in Lemma 4.1, not even for resolution V fractions. Godolphin (2019) applied a search for such $\mathbf{X}$ matrices as follows:

- Fix the first column of $\mathbf{X}_\mathrm{I}$, and inspect all $(2^q-1)^{n-p-1}$ possible picks of $n-p-1$ columns from $\mathcal{X}_q$ for the remaining positions of $\mathbf{X}_\mathrm{I}$.
- For each $\mathbf{X}_\mathrm{I}$, obtain $\mathbf{X}_\mathrm{II} = \mathbf{X}_\mathrm{I}\mathbf{Z}^\top$, and inspect it for all-zero columns. Discard the case, if an all-zero column is found.
- If $\mathbf{X}_\mathrm{II}$ is acceptable, obtain $\mathbf{X} = (\mathbf{X}_\mathrm{I} \vdots \mathbf{X}_\mathrm{II})$ and inspect the profile it implies.

A slightly more efficient approach is implemented in 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 also restrict the cth column to at most c choices, so that the number of combinations to be checked can be reduced from $(2^q-1)^{n-p-1}$ to $(n-p)!$ for $n-p \leq 2^q-1$ and to $(2^q-1)!(2^q-1)^{n-p+1-2^q}$ otherwise. Choices with fewer than q different columns in $\mathbf{X}$ are discarded as well as choices with all-zero columns in $\mathbf{X}_\mathrm{II}$.

Example 4, continued: For blocking the fraction 13–5.1 into blocks of size 4, $3! \cdot 3^5 = 1458$ $\mathbf{X}_\mathrm{I}$ matrices need to be inspected. The automated search of package **FrF2** yields

$$\mathbf{X} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{9}$$

Blocking a design with this $\mathbf{X}$ matrix aliases all within part 2fis for the partition {(A,E,F,G,H), (B,J,K,L,N), (C,D,M)} with the block factor and keeps all between part 2fis orthogonal to block factor effects. As the fraction has resolution V, all the 2fis not confounded by the block factor are clear, so that the number of clear 2fis can still be calculated like for blocked full factorial designs. As the $\mathbf{X}$ matrix implies the profile <5,5,3>, there are $5 \cdot 5 + 5 \cdot 3 + 5 \cdot 3 = 55$ clear 2fis (one less than in the most balanced profile <5,4,4>, which is not compatible with the confounding structure of the fraction). As an aside, note that the only profiles compatible with the fraction are <5,5,3>, <7,3,3>, <7,5,1> and <9,3,1>. The catalogue contains a single fraction only for 256 run designs. Example 10 in Appendix D provides a custom set of generating contrasts for which a larger variety of profiles can be obtained, among them the most balanced profile.

For fractions of resolution V, obtaining $\mathbf{X}$ is the only complication: since the fraction keeps all 2fis clear, any confounding of 2fis arises as confounding with the block factor. For resolution IV fractions, there are already some non-clear 2fis to start with: all pairs of factors that occur in at least one defining word of length 4 are not clear. According to Lemma 4.1, the number of clear 2fis for the blocked fraction is at most the minimum of (4) and the number of clear 2fis from the unblocked fraction.

Example 5: A 32 run design in 7 factors is to be blocked into blocks of size 4. A full factorial in 7 factors allows 16 clear 2fis (see Example 3), i.e. the blocked full factorial confounds 5 2fis. The unblocked fraction 7–2.1 has 15 clear 2fis; the best blocked fraction has 12 clear 2fis, i.e. 3 less than the upper bound obtained from blocked full factorial and unblocked fraction. The $\mathbf{X}$ matrix has the profile <3,2,2>, but the number of clear 2fis can of course no longer be calculated from this information. The fraction 7–2.1 is the only resolution IV fraction for seven factors in 32 runs. Thus, it is not possible to keep more than 12 2fis clear when blocking a resolution IV fraction into blocks of size 4.

Example 6: Blocking the fraction 13–6.1 into blocks of size 4, the maximum number of achievable clear 2fis is 52. The automatically created design is obtained with

$$\mathbf{X} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{10}$$

(profile <5,4,4>) and has the 52 clear 2fis AC, AD, AF, AH, AJ, AK, AM, BC, BD, BE, BF, BG, BK, BM, CE, CG, CH, CJ, CL, CN, DE, DG, DH, DJ, DL, DN, EF, EH, EJ, EK, EL, EM, FG, FH, FJ, FL,

Table 3: Profiles obtainable from four 128 run fractions for 13 factors, when blocked into blocks of size 4

13–6.1			13–6.2			13–6.3			13–6.4		
3	3	7	2	3	8	1	4	8	2	3	8
3	4	6	2	4	7	1	6	6	2	4	7
3	5	5	2	5	6	4	4	5	2	5	6
4	4	5	3	3	7	.	.	.	3	3	7
.	.	.	3	4	6	.	.	.	3	4	6
.	.	.	3	5	5	.	.	.	3	5	5
.	.	.	4	4	5	.	.	.	4	4	5

FN, GH, GJ, GK, GL, GM, HK, HM, HN, JK, JM, JN, KL, KN, LM, MN. The full factorial blocked with this **X** matrix would keep 56 2fis clear. Twelve 2fis of the unblocked fraction are not clear, among them AB, AL, BN and LN, which would have been clear in a blocked full factorial with the **X** from (10). Thus, the estimability CIG of the blocked fraction is slightly worse than that of a blocked full factorial. It has been checked (see the code in the appendix) that the number of clear 2fis cannot be improved by using a different resolution IV fraction.

4.2 Estimability requirements

The following lemma states necessary conditions for blocking while keeping specified 2fis clear:

Lemma 4.2. *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) The fraction $\mathcal{F}$ keeps the requirement CIG G clear, i.e. the estimability CIG of $\mathcal{F}$ contains a subgraph that is isomorphic to G.
(ii) The requirement CIG 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 the requirement CIG.

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 3.2, 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.

Condition (iii) of the lemma is a real constraint. This is demonstrated by inspecting the profiles that can be obtained from blocking the four best 128 run fractions for 13 factors into blocks of size 4 (13–6.1 with 66 clear 2fis, 13–6.2 with 66 clear 2fis, 13–6.3 with 60 clear 2fis and 13–6.4 with 60 clear 2fis): Table 3 shows their feasible profiles. The MA fraction allows fewer profiles than 13–6.2 and 13–6.4, but more than 13–6.3. Fraction 13–6.3 is the only one with a profile that contains a "1" element, i.e. that can accommodate a requirement CIG with *all* 2fis of a selected factor clear. All fractions offer the maximally balanced profile <5,4,4>, which only confounds 22 2fis with blocks in a full factorial, but confounds between 26 and 34 2fis in these four fractions (see Example 6 for the best one). The third fraction offers the fewest profiles, among them however two that cannot be produced by any of the other three fractions. Note that a full factorial would have 14 possible profiles that involve all three vectors of $\mathcal{X}_2$; five of these do not occur for any of the four fractions; it has been verified that these can be obtained from other catalogued resolution IV fractions. For blocks of size 8 (not in the table), the fraction 13–6.4 yields all possible 7-profiles. The fractions with smaller aberration cannot produce all of these: fractions 13–6.1 to 13–6.3 cannot obtain the <7,1,1,1,1,1,1> and <4,4,1,1,1,1,1> profiles, fraction 13–6.1 additionally cannot obtain the most balanced profile <2,2,2,2,2,2,1>.

According to Lemma 4.1, 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 either regard this 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.

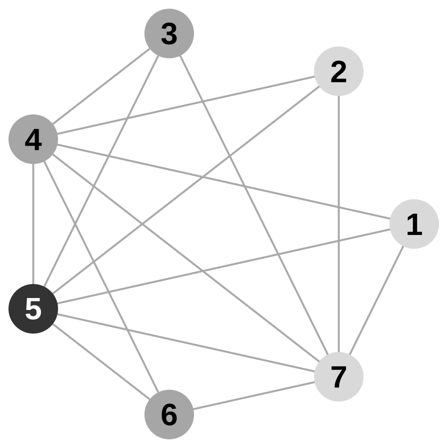

Figure 4: Estimability CIG of the unblocked MA fraction 7-2.1, coloured in terms of the successful mapping for Example 7 (this is of course no proper colouring for this CIG). The generating contrasts for the fraction are: factor 6 from the 3-factor interaction of factors 1, 2, and 3 and factor 7 from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

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, and the templates also list 2fis that are confounded in the fraction although they would be unconfounded in the blocked full factorial. Based on the templates, it is then 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, fractions are pre-filtered for compatibility with the requirement CIG. Blocking is attempted only for those fractions whose unblocked version can accommodate the requirement CIG; these are identified with the algorithm described in Grömping 2012. Subsequently, for every **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 7: Like in Example 3, 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 4, coloured according to the successful mapping found by the algorithm. All the necessary conditions of Lemma 4.2 are satisfied; for condition (iii), see Example 5, where a blocking of fraction 7–2.1 with profile <3,2,2> was obtained, which corresponds to the left-hand side colouring of the requirement CIG in Figure 1. 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 **X** matrix that corresponds to the profile <3,3,1>, i.e. the profile of the right-hand side graph in Figure 1, with the partition {{A,D,E}, {B,F,G}, {C}}. With this partition and **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 **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:

$$\mathbf{X}_\mathrm{I} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \qquad (11)$$

Note that the column order of the matrix **Z** is not remapped but corresponds to the original factor order in the fraction 7–2.1.

Example 8: 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 only be found because a subgraph isomorphism check is done for each **X** matrix during the search process (see code in the appendix).

Example 9: 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 4.2 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 4.2 is fulfilled for these fractions). For condition (iii) of Lemma 4.2, 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". Thus, as can be seen from Table 3, the fraction 13–6.3 has MA among fractions that fulfil the necessary condition (iii) of Lemma 4.2. 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 (see Table 3). A search over all 197 blocked fractions whose estimability CIG contains the requirement CIG showed that a few more 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 behavior or the fraction 13–6.16 with more clear 2fis.

Note that the successful fraction 13–6.3 of Example 9 is *dominated*, i.e. its estimability CIG is a subgraph of the estimability CIG of a fraction with less aberration. Based on a proposal by Wu, Mee and Tang (2012), Grömping (2014b) improved the algorithm of Grömping (2012) by restricting the search to dominating fractions, i.e. fractions that are not dominated. This brought about a substantial efficiency gain for larger problems. Consequently, many 128 run dominated fractions, among them 13–6.3, were excluded from the standard catalogue `catlg` of package **FrF2**. For the blocking application, Example 9 and Table 3 showed that a dominated fraction can yield the best solution. Luckily, larger catalogues including dominated fractions are available in the auxiliary package **FrF2.catlg128**; these should be used instead of the standard catalogue for blocking 128 run fractions (see also the code for Example 9 in Appendix C).

5 The algorithm and its implementation in 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 5 and 6 give a high-level overview of the algorithm implemented for this purpose: Figure 5 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 6 is applied. In 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 6 can be followed with re-entering step 3 of Figure 5, eliminating the failed candidate from the eligible fractions. At present, this restart has to be manually triggered.

5.1 Full factorial without a requirement CIG

The automatic algorithm will quickly return a blocked design with $\phi_{\max}$ clear 2fis. The factor partition will place the first $\min(2^q-1, n)$ factors in separate parts (let us number them as $1, \dots, 2^q-1$). For more than 2^q-1 parts, 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 R function `FF_from_X`.

Failure to create a blocked design may be caused by resource problems ($N = 2^n$ becomes large very fast).

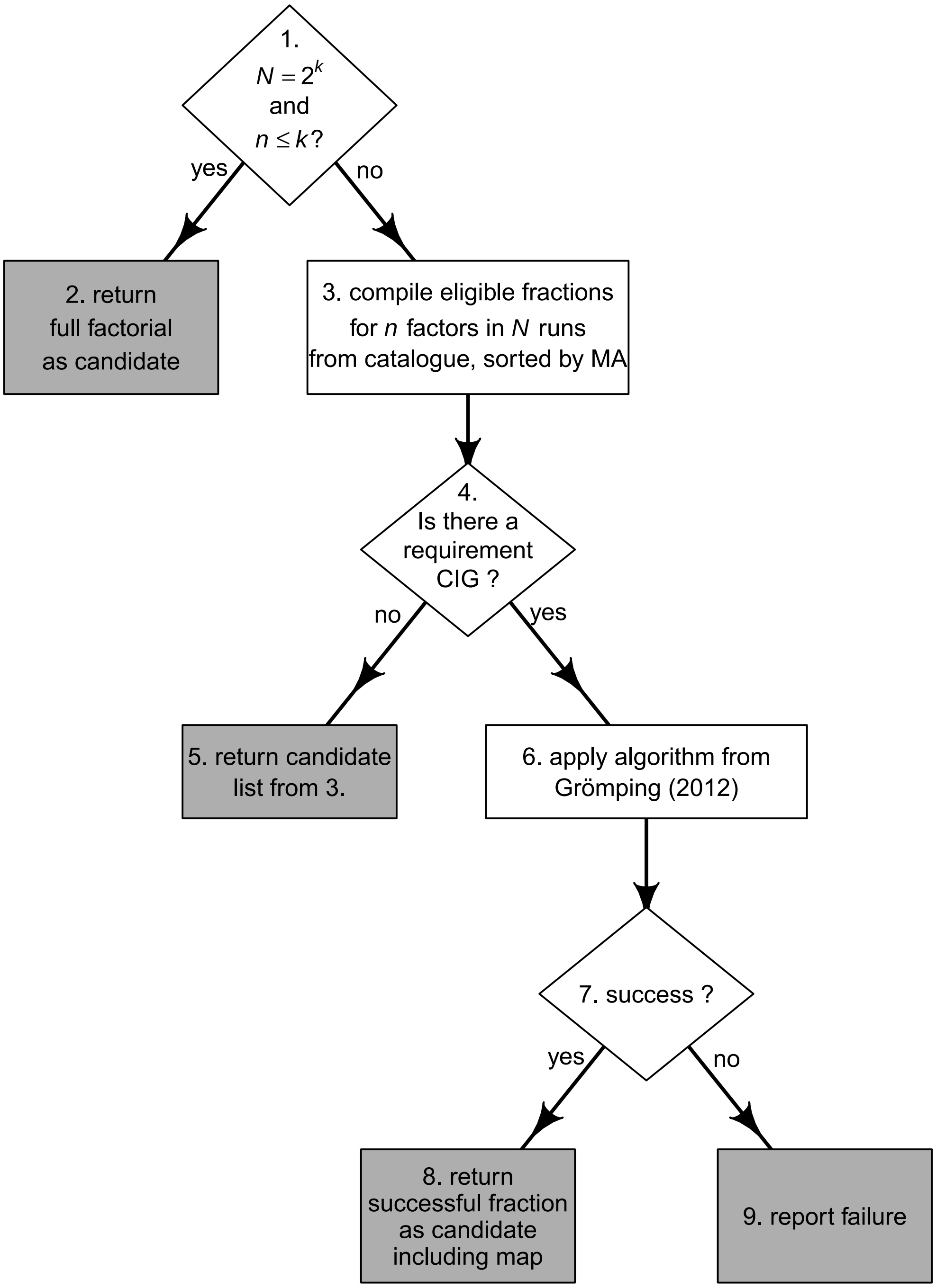

Figure 5: The algorithm for selecting one or more candidate fraction(s), given n, N, q, and possibly a requirement CIG

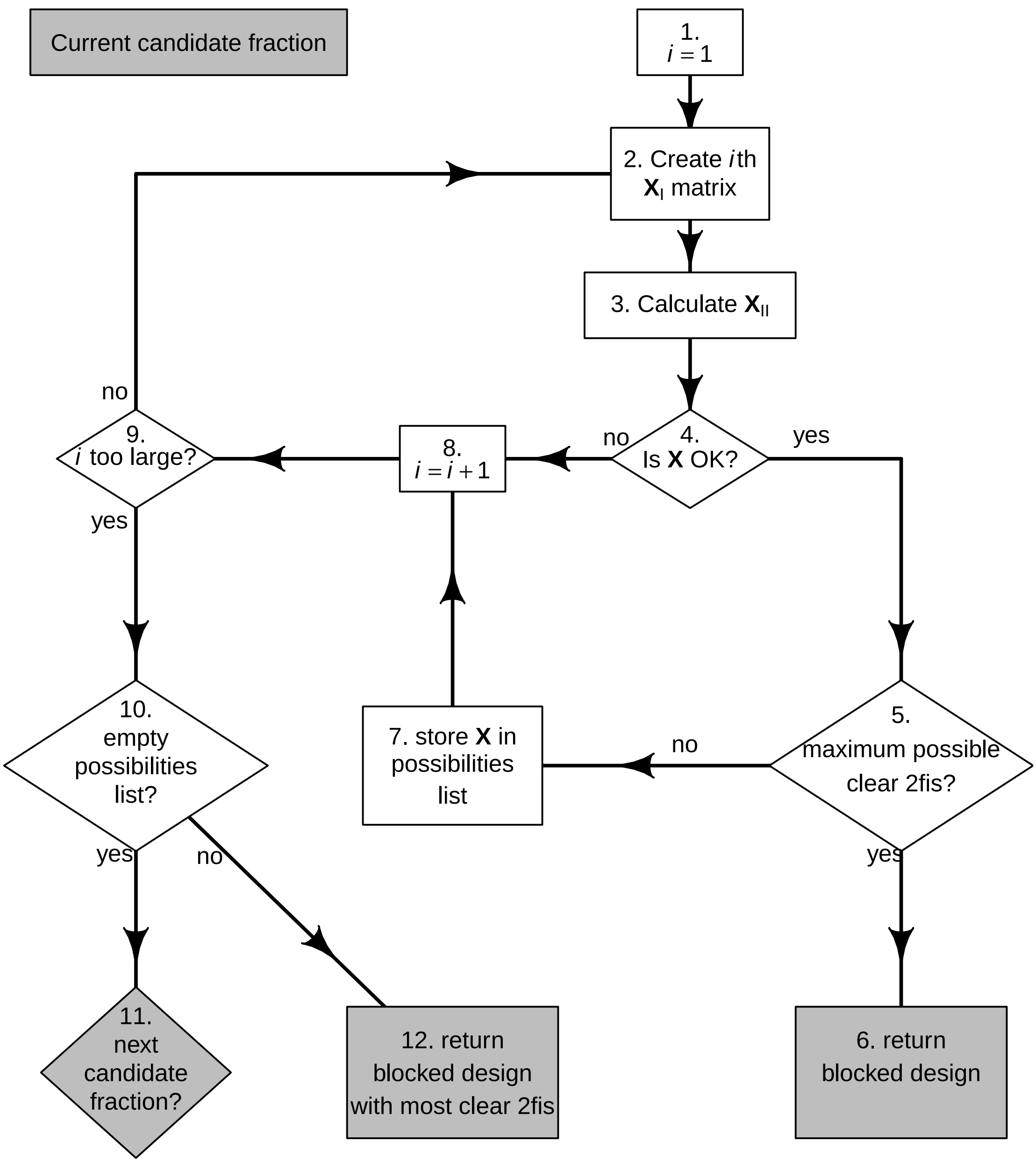

Figure 6: The algorithm for blocking a candidate fraction. For keeping the diagram simple, the check in step 4. includes the check for the requirement CIG, if applicable. This check involves a subgraph isomorphism check whether the requirement CIG is a subgraph of the estimability CIG of the blocked design.

5.2 Full factorial with a requirement CIG

The automatic algorithm starts from the default without a requirement CIG, and afterwards loops through **X** column choices according to the approach described in Section 4.1, with the first components moving fastest. For each **X**, a subgraph isomorphism check will indicate whether the requirement CIG can be accommodated in the estimability CIG. In 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 3).

The process may be slow, if the requirement CIG is such that early **X** choices cannot accommodate the blocking, or (of course) if the requirement CIG cannot be accommodated in blocks of size 2^q. Especially for large full factorials and requirement CIGs that are easily understood, it may thus be advisable to manually create a suitable **X** matrix with desired properties, e.g. using R function `X_from_parts`.

The algorithm fails if the requirement CIG has a chromatic number larger than $2^q - 1$, or for resource reasons.

5.3 Fractional factorial without a requirement CIG

For $k = n - p$, the sequence of $\mathbf{X}_\mathrm{I}$ matrices is obtained in the same way as the sequence of **X** matrices for the full factorial case. **X** matrices for which $\mathbf{X}_\mathrm{II}$ contains any all-zero columns are discarded, as are **X** matrices with 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 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 6 in Appendix C exemplifies how to check, whether the number of clear 2fis can be improved.

The algorithm will not fail, as long as $n \leq 2^{n-p-1}$ (i.e., a resolution IV fraction exists), except for resource reasons. However, for too large p or too small q, there may be no or too few clear 2fis.

5.4 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 5). Subsequently, the algorithm loops through $\mathbf{X}_\mathrm{I}$ matrices in the same way as in Section 5.3, with an additional subgraph isomorphism check whether the blocked design obtained with the **X** matrix can accommodate the requirement CIG. For resolution IV fractions, a separate subgraph isomorphism check for each **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 5. 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 5 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 **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.

According to Lemmata 4.1 and 4.2, 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 5 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 **X** matrix without all-zero columns cannot be found based on the current candidate fraction that was returned by the algorithm of Figure 5.

Failures of type (i) produce an error message that starts with `The required interactions cannot be accomodated 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; for example, for 256 runs, the built-in catalogue only contains MA fractions n–p.1).

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 function `CIG` from package **FrF2**, ideally in interactive mode). (There is currently no reliable algorithm in R that automates this task.) If the requirement CIG is $2^q - 1$-colourable, the failure must be of type (iii).

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 5 can be started again after removing the failed fraction from the eligible fractions in step 3. We already met an example of this kind as Example 9 with 13 factors in 128 runs in blocks of size 4. In this example, 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. Where such a restart is necessary, it is useful to run step 6 of the algorithm of Figure 5 without the efficiency improvement of Grömping (2014b), because dominated fractions can be very promising for the blocking task (e.g. 13–6.3).

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 function `FrF2`: Appendix C shows code for Example 9 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 `catlg128.5to18` from that package.

6 Discussion

Godolphin (2019) proposed a powerful tool that allows to combine blocking a full factorial or a fraction into small blocks 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 obtaining blocks of size 4, from which practitioners can work out how to accommodate their specific needs. The algorithm presented in this paper automates the process, using a sorted catalogue of fractions, which stores the estimability CIG with each fraction. If a requirement CIG is specified, the proposed algorithm uses the algorithm by Grömping (2012) for selecting a candidate fraction. 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.

For the algorithm of this paper, it is advisable to conduct the algorithm by Grömping (2012) without the efficiency improvement from Grömping (2014b): the restriction to dominating fractions that was proposed in Grömping (2014b) can deteriorate the result, as was for example seen in Example 9, where the most successful fraction was a dominated one.

The proposed algorithm relies on the availability of a catalogue of fractions. For 256 runs and more, the catalogue in R package **FrF2** so far contains the MA fractions n–p.1 only. This 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 and more runs – e.g. the ones available from Hongquan Xu's website in support of his 2009 paper – will be made available for R package **FrF2** (possibly in another separate auxiliary package), for supporting the blocking methodology. While these are not yet available, hand-picked elements from external catalogues can be used; these need to be cast as a catalogue object (see Example 10 in Appendix D).

For unblocked fractions, MA is universally accepted as a suitable quality criterion. The fraction, on which a blocked design from the algorithm proposed in this paper is based, has MA for the treatment factors among all fractions that fulfil the requirements (block size and possibly a requirement CIG) and keeps 2fis clear as much as possible within the successful fraction. Quality criteria have also been proposed for blocked designs (see Godolphin 2019 and 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 blocked 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.

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 a split-plot 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 **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.

Bollobás, B. and Guy, R.K. (1983). *Resolution IV designs with 128 runs. Journal of Quality Technology* **37**, 282-293.

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

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.

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.

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/.

Appendix

A: Fractions used in this paper

The following fractions have been used in Examples 4 to 9, and Tables 1 and 3.

```
catlg["6-2.1"]   ## Table 1
```

```
## Design:  6-2.1
##    16  runs,  6  factors,
##    Resolution  IV
##    Generating columns:  7 11
##    WLP (3plus):  0 3 0 0 0 ,  0  clear 2fis
```

```
catlg["13-5.1"]  ## Example 4
```

```
## Design:  13-5.1
##    256  runs,  13  factors,
##    Resolution  V
##    Generating columns:  127 143 179 213 105
##    WLP (3plus):  0 0 3 12 12 ,  78  clear 2fis
```

```
## Table 3 and Examples 6 and 9
catlg128.8to15[c("13-6.1", "13-6.2", "13-6.3", "13-6.4", "13-6.12", "13-6.16")]
```

```
## Design:  13-6.1
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 85 44 86
##    WLP (3plus):  0 2 16 18 ,  66  clear 2fis
##  Factors with all 2fis clear:  E G H J K
## Design:  13-6.2
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 85 46 61
##    WLP (3plus):  0 2 16 20 ,  66  clear 2fis
##  Factors with all 2fis clear:  D E G J L
## Design:  13-6.3
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 49 74 124
##    WLP (3plus):  0 3 12 24 ,  60  clear 2fis
##  Factors with all 2fis clear:  K
## Design:  13-6.4
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 85 44 82
##    WLP (3plus):  0 3 14 17 ,  60  clear 2fis
##  Factors with all 2fis clear:  A H J
## Design:  13-6.12
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 81 44 82
##    WLP (3plus):  0 4 10 22 ,  57  clear 2fis
##  Factors with all 2fis clear:  H J K
## Design:  13-6.16
##    128  runs,  13  factors,
##    Resolution  IV
##    Generating columns:  31 103 43 49 74 62
##    WLP (3plus):  0 4 12 22 ,  57  clear 2fis
```

```
##  Factors with all 2fis clear:  C J K
```

```r
catlg["7-2.1"]  ## Examples 5 and 7
```

```
## Design:  7-2.1
##    32  runs,  7  factors,
##    Resolution  IV
##    Generating columns:  7 27
##    WLP (3plus):  0 1 2 0 0 ,  15  clear 2fis
##  Factors with all 2fis clear:  D E G
```

```r
catlg["9-3.1"]  ## Example 8
```

```
## Design:  9-3.1
##    64  runs,  9  factors,
##    Resolution  IV
##    Generating columns:  7 27 45
##    WLP (3plus):  0 1 4 2 0 ,  30  clear 2fis
##  Factors with all 2fis clear:  D E F H J
```

B: Relevant function arguments, functions and objects in R package FrF2

The most elementary functions (see Example 3) are

- `X_from_parts` for the creation of an **X** matrix from a partition (=list of parts) and
- `FF_from_X` for the creation of a blocked full factorial from an **X** matrix.

These light-weight functions can be used for manual handling of full factorials. Function `FF_from_X` returns a proper class `design` object, like all design-generating functions from the package suite to which **FrF2** belongs.

For more heavy-weight implementations involving fractions and requirement CIGs, function `FrF2` implements the entire design creation in an automated way. Its most relevant arguments are

- the number of runs `nruns`
- the number of factors `nfactors`
- optional: `design` for the fraction to be blocked (name of catalogue element; in that case, `nruns` and `nfactors` are not needed)
- optional: `generators` for the fraction to be blocked (cannot be combined with the `estimable` functionality)
- optional: a vector of factor names `factor.names` (not necessary, but very helpful in case factors are permuted)
- optional: a number of blocks or other ways of specifying block generation (e.g. block generators), `blocks`
- optional: `alias.block.2fis=TRUE` for permitting the block effect to be aliased with 2fis
- optional: `block.name`
- optional: a specification of 2fis to be kept clear, `estimable` (assuming that the default argument `clear=TRUE` is not changed)

The internal functions `mapcalc` and `mapcalc.block` are the workhorse functions behind the estimability functionality of function `FrF2`. Step 6 of the algorithm's candidate search portion uses function `mapcalc` for identifying candidate fractions that can accommodate the requirement CIG, and a suitable map for them. Function `mapcalc.block` is used in Step~4 of the algorithm of Figure~6; its use can be deactivated for resolution IV fractions (`useV=FALSE` argument); in that case, the map returned by function `mapcalc` is used throughout the algorithm of Figure 6, see Example 8). Further functions can support users who want to closely inspect some details or want to try to improve the number of clear 2fis after a successful run of `FrF2`:

- `phimax` calculates the upper bound from formula (3) or the number (4).
- `colpick` is the work horse function behind the algorithm of Figure 6: it returns the **X** matrix which is then post-processed by function `FrF2` in the automated blocking process. Optionally, the function can be requested to return not only the most successful **X** matrix but all successful **X** matrices for

a given candidate fraction; this can e.g. be useful for making sure the number of clear 2fis is as large as possible. The function does not only return the **X** matrix but also the resulting set of clear 2fis, and possibly a mapping that indicates how to permute factor names in order to ascertain a requested estimability pattern (element `map`, see Example 7 for detail). Caution: When requesting all successful matrices, or where the blocking request cannot be satisfied, the function may take a long time, whenever $k = n - p$ and/or q is large.

- The internal function `colpickIV` is the work horse function behind the algorithm of Figure~6 for resolution IV fractions, if a user decides to switch off the subgraph isomorphism check for each **X** (`useV=FALSE` in function `FrF2`). The function uses the initial map from step 6 of Figure 5 throughout.
- `blockgencreate` calculates block generators from an **X** matrix and design generating information. It can be used for creating a suitable `blocks` argument in calls to function `FrF2`, if one wants to manually block a design with a self-selected **X** matrix.
- `CIG` depicts a requirement or estimability `CIG` (or any other graph that can be represented as an edge list in a form understood by the function). It can be used for manually inspecting a requirement CIG for its chromatic number or a suitable partition.
- `clear.2fis` extracts the clear 2fis (in matrix form) from a `catlg` object, `nclear.cfis` the number of clear 2fis.
- `generators` extracts the generating contrasts in human-readable form from a `catlg` object.
- `X_from_parts` provides an **X** matrix from a partition (used for Example 3);
- `X_from_profile` provides an **X** matrix from a profile; its columns are sorted by part size (largest part in front).
- `Xcalc` calculates an **X** matrix from an $\mathbf{X}_\mathrm{I}$ matrix together with a generator or a `catlg` object. This function is of academic interest; it will usually not be needed for design creation.

The following objects support their use:

- `catlg` is the standard catalogue of fractional factorial 2-level designs.
- With package **FrF2.catlg128**, an extensive set of further 128 run catalogues in different numbers of runs are provided. These help, for example, in finding the clear design of Example 9.
- `Yates` is a list of Yates matrix column vectors (e.g. `c(1,2,3)` for ABC) in the positions where they belong for Yates order (see Table 1).
- `Letters` is a vector of upper and lower case letters for factor names; the letters “I” and “i” are missing from the vector, because their use as factor names can be confusing.

C: R code for examples

Examples 1 to 6

Example 1 used a naïve **X** matrix for blocking a full factorial design with seven factors into blocks of size 8.

```
## Example 1
X <- rbind(
  c(1,0,0,1,1,1,1),
  c(0,1,1,0,1,0,1),
  c(0,1,1,0,0,1,1)
)
## all effects that are aliased with blocks
FrF2:::blockgengroup(X)
## block generators
blockgencreate(X)
## create the design
FF_from_X(X)
```

Example 2 used an optimal **X** matrix for blocking a full factorial design with seven factors into blocks of size 8. `colpick(7,3)$X` would find a different such matrix.

```r
## Example 2
X <- rbind(
  c(1,0,0,0,1,1,1),
  c(0,0,1,1,1,0,1),
  c(0,1,1,0,0,1,1)
)
## all effects that are aliased with blocks
FrF2:::blockgengroup(X)
## block generators
blockgencreate(X)
## create the design
FF_from_X(X)
```

Example 3 manually created a suitable **X** matrix for blocking a full factorial in seven factors such that the requirement CIG of Figure 1 is kept clear. The code for doing this with package **FrF2**:

```r
X <- X_from_parts(7,2,list(c("A","D","F"), c("B","G"), c("C","E")))
summary(FF_from_X(X))

## function FrF2 also yields such a design (a different but isomorphic one)
summary(FrF2(128, 7, factor.names=Letters[15:21],
      estimable=requ, alias.block.2fis = TRUE, blocks=32))
```

Example 4 initially used a naïve $\mathbf{X}_{\mathrm{I}}$ matrix for blocking the MA fraction 13–5.1 into blocks of size 4.

```r
# J="ABCDEFG", K="ABCDH", L="ABEFH", M="ACEGH", N="ADFG")
Z <- rbind(
     c(1, 1, 1, 1, 1, 1, 1, 0),
     c(1, 1, 1, 1, 0, 0, 0, 1),
     c(1, 1, 0, 0, 1, 1, 0, 1),
     c(1, 0, 1, 0, 1, 0, 1, 1),
     c(1, 0, 0, 1, 0, 1, 1, 0)
     )
XI <- rbind(
  c(1,1,1,1,1,0,0,0),
  c(0,0,0,1,1,1,1,1)
)
XII <- (XI%*%t(Z))%%2
X <- cbind(XI, XII)
```

A better **X** matrix was obtained using the search algorithm of function `colpick`.

```r
X <- colpick("13-5.1", 2)$X
```

Example 5 illustrated the numbers of clear 2fis for the unblocked fraction 7–2.1, the blocked full factorial in seven factors, and the blocked fraction 7–2.1.

```r
## unblocked fraction
nclear.2fis(catlg["7-2.1"])
## blocked full factorial with blocks of size 4
phimax(7, 2)
## blocked fraction
length((coll <- colpick("7-2.1", 2))$clear2fis)
## inspect profile
coll$X
```

Example 6 inspected blocking the resolution IV fraction 13–6.1 into blocks of size 4, without a requirement CIG .

```r
## find best X matrix for blocking fraction, block size 2^2
col_ex6 <- colpick("13-6.1", 2)
```

```
## number of clear 2fis
 length(col_ex6$clear2fis)
## the X matrix
 col_ex6$X
## the design (32 blocks of size 4)
 desEx6 <- FrF2(128, 13, design="13-6.1", blocks=32, alias.block.2fis = TRUE)
## number of confounded 2fis of the fraction
 choose(13,2) - nclear.2fis(catlg["13-6.1"])
## words of length 4 of the fraction
 FrF2:::words.all(7, catlg[["13-6.1"]]$gen, max.length = 4)
## two disjoint words, implying six non-clear 2fis each
## the word ABLN (1,2,11,13) causes the confounding of
##      clear 2fis from the blocked full factorial
```

Function FrF2 would also have used the fraction 13–6.1 without specifying it in the design argument. There are four less clear 2fis than for the full factorial. One might have attempted an improvement with the code below, but none was found.

```
nn2 <- names(catlg128.8to15[nruns(catlg128.8to15)==128 &
                              nfac(catlg128.8to15)==13])
target <- phimax(13,2)
nccur <- 0
fraccur <- character(0)
system.time(for (nam in nn2){
  #print(nam)
  suppressMessages({
    now <-
      length(colpick(nn2, 2, select.catlg=catlg128.8to15)$clear2fis)
    if (now > nccur) {
      nccur <- now
      fraccur <- nam
    }
     if (now==target) break
  })
})
nccur
```

Examples 7 and 8

Example 7 refers to the requirement CIG of Figure 1, which was created with the code below.

```
requ <- c("AB","AC","BC","BD","BE","CD","CF","CG","EF","EG")
par(mfrow=c(1,2), mar=c(0,0,0,0))
## ADF CE BG

CIG(requ, vertex.size=30,
    vertex.color=c(light, middle, dark, light, dark, light, middle),
    vertex.frame.color=c(light, middle, dark, light, dark, light, middle),
    vertex.label.color=c("black","black","white","black","white",
                         "black","black"),
    vertex.label.font=2, vertex.label.family="sans",
    static=TRUE, layout=igraph::layout_as_tree, root=3)
## ADE BFG, C
CIG(requ, vertex.size=30,
    vertex.color=c(light, middle, dark, light,light, middle, middle),
    vertex.frame.color=c(light, middle, dark, light,light, middle, middle),
    vertex.label.color=c("black","black","white","black","black",
                         "black","black"),
```

```
    vertex.label.font=2, vertex.label.family="sans",
    static=TRUE, layout=igraph::layout_as_tree, root=3)
```

The **X** matrix and column mapping was produced with the command

```
coll <- colpick("7-2.1", 2, estimable=requ)
```

The code below illustrates how to verify that this indeed yields an appropriate solution.

```
## Z-matrix for the unblocked fraction
## original column order
Z <- t(sapply(catlg[["7-2.1"]]$gen,
     function(obj) rev(sfsmisc::digitsBase(obj, 2, 5))))

## coll$X implies the profile 3,3,1
## i.e. partition {{A,D,E}, {B,F,G}, {C}}

## the map element of coll implies
## column allocations
# A:1, B:4, C:7, D:2, E:5, F:3, G:6

# A(1) in the same part with D(2) and E(5), columns 1, 2 and 5
# B(4) in the same part with F(3) and G(6), columns 3 and 4
# C(7) alone in a part, column 7

## reproduce
XI.4_RHS <- rbind(c(0,0,1,1,0), ## RHS
                  c(1,1,0,0,1))
X.4_RHS <- cbind(XI.4_RHS, (XI.4_RHS%*%t(Z))%%2)
## additional column respect the profile
```

For Example 8, blocking the fraction 9–3.1 is successful but would fail when suppressing subgraph isomorphism search for each **X** (useV=FALSE).

```
requEx8 <- compromise(9,8:9, class=4)$requirement
fn <- c(paste0("C",1:7), paste0("N", 1:2))
## suppressing subgraph isomorphism checking for individual
## X matrices makes the algorithm fail
## (useV=FALSE does that)
nosuccess <- FrF2(64,9, blocks=16, estimable=requEx8,
                  factor.names=fn, alias.block.2fis = TRUE,
                  useV=FALSE)
## the default with subgraph isomorphism checks is successful
des64blocked4 <- FrF2(64,9, blocks=16, estimable=requEx8,
                  factor.names=fn, alias.block.2fis = TRUE,
                useV=TRUE)
```

Example 9

This example illustrates how to handle cases for which the default attempt is unsuccessful. The MA fraction accommodates the requirement CIG, but cannot be blocked with an appropriate profile. Solutions can be searched for by looping over potential candidate fractions. For less complicated problems, it would also be possible to manually remove the failed fraction(s) from `select.catlg`, e.g. with `select.catlg=catlg[-(1:which(names(catlg)=="13-6.1"))]`.

Within catalogue `catlg`, the 11th element 13–6.12 is the first to accommodate the requirement CIG with the requested blocking. A better design, based on the fraction 13–6.3, can be found from searching within the larger catalogue `catlg128.8to15` from R package **FrF2.catlg128**. Code for both searches is shown below.

```
requEx9 <- compromise(13,1)$requirement

## no success
nosuccess <- FrF2(128,13, blocks=32, factor.names=Letters[15:27],
                  estimable=requEx9, alias.block.2fis = TRUE)

## loop through designs from catlg
nn1 <- names(catlg[nruns(catlg)==128 & nfac(catlg)==13])
for (nam in nn1){
  suppressMessages(
    des128fromcatlg <- try(
         FrF2(design=nam, blocks=32,
            factor.names=Letters[15:27], estimable=requEx9,
            alias.block.2fis = TRUE),
        silent=TRUE)
    )
  ## stop at first success
  if (!"try-error" %in% class(des128fromcatlg)) break
}
## found 13-6.12
## worse aberration than the optimum possible
## see below
design.info(des128fromcatlg)
## has 36 clear 2fis (78 - 34 aliased with block - 8 aliased)

## loop through designs from catlg128.8to15
nn2 <- names(catlg128.8to15[nruns(catlg128.8to15)==128 &
                               nfac(catlg128.8to15)==13])
for (nam in nn2){
  suppressMessages(
    des128fromcatlg128 <- try(
         FrF2(design=nam, blocks=32,
           factor.names=Letters[15:27], estimable=requEx9,
           select.catlg = catlg128.8to15,
           alias.block.2fis = TRUE),
        silent=TRUE)
    )
  ## stop at first success
  if (!"try-error" %in% class(des128fromcatlg128)) break
}
design.info(des128fromcatlg128)
## has 48 2fis unconfounded with blocks (78 - 30 - 12)
## 36 2fis clear

## checking whether there is a design with more clear 2fis
ncV <- suppressMessages(
  sapply(nn2, function(obj){
     hilf <- try(length(colpick(obj, 2, select.catlg = catlg128.8to15,
                    estimable=requEx9)$clear2fis), silent=TRUE)
     if ("try-error" %in% class(hilf)) return(NA) else return(hilf)
  })
  )
table(ncV, useNA = "ifany")
which.max(ncV)   ## 13-6.16, 40 clear 2fis
ncV[ncV>=36]
```

D: Worked Example 10

The code below casts the resolution V fraction 13–5.2 from Hongquan Xu's website into a class `catlg` object, for use in functions `colpick` or `FrF2`. The fraction has more and entirely different profiles than fraction 13–5.1 (which has profiles <5,5,3>, <7,3,3>, <7,5,1> and <9,3,1>); it thus is compatible with different requirement CIGs. The resulting blocked design has 56 clear 2fis, using the most balanced profile.

```
gen <- c(127,143, 179, 85, 150)  ## from Xu website
names(Yates)[gen]
```

```
## [1] "ABCDEFG" "ABCDH"   "ABEFH"   "ACEG"    "BCEH"
```

```
## prepare named list
##    (for self-made generator, could use x instead of 2
##     in the name),
## WLP entries start with length 1
## nclear.2fis and clear.2fis for resolution V and higher
##     are trivial
## dominating for resolution V and higher is FALSE
##     for non MA fractions
mycatlg <- list("13-5.2"=list(nruns=256, nfac=13, gen=gen,
                WLP=c(0,0,0,0,5,10,10),
                res=5, nclear.2fis=choose(13,2),
                clear.2fis=combn(13,2),
                dominating=FALSE))
## assign class catlg to make it usable for the functions
class(mycatlg) <- c("catlg", "list")
colll <- colpick("13-5.2", 2, all=TRUE, select.catlg = mycatlg)
```

```
## checking up to 1458 matrices
```

```
unique(colll$profiles)
```

```
## [[1]]
## [1] 2 3 8
##
## [[2]]
## [1] 2 2 9
##
## [[3]]
## [1] 3 4 6
##
## [[4]]
## [1] 2 5 6
##
## [[5]]
## [1] 2 4 7
##
## [[6]]
## [1] 4 4 5
```

```
summary(descustom <- FrF2(256, 13, blocks=64, select.catlg = mycatlg, alias.block.2fis = TRUE))
```

```
## checking up to 1458 matrices
```

```
## Call:
## FrF2(256, 13, blocks = 64, select.catlg = mycatlg, alias.block.2fis = TRUE)
##
## Experimental design of type  FrF2.blocked
## 256  runs
## blocked design with  64  blocks of size  4
```

```
##
## Factor settings (scale ends):
##    A  B  C  D  E  F  G  H  J  K  L  M  N
## 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
## 2  1  1  1  1  1  1  1  1  1  1  1  1  1
##
## Design generating information:
## $legend
##  [1] A=A B=B C=C D=D E=E F=F G=G H=H J=J K=K L=L M=M N=N
##
## $`generators for design itself`
## [1] J=ABCDEFG K=ABCDH   L=ABEFH   M=ACEG    N=BCEH
##
## $`block generators`
## [1] BC  ABD ABE BF  AG  AH
##
##
## no aliasing of main effects or 2fis  among experimental factors
##
## Aliased with block main effects:
##  [1] AG AH AM BC BF BJ BN CF CJ CN DE DK DL EK EL FJ FN GH GM HM JN KL
```