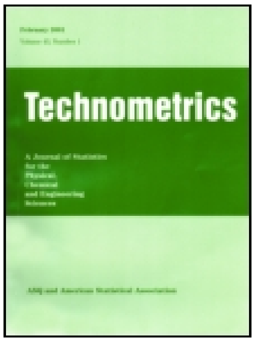
Constructing two-level designs by concatenation of strength-3 orthogonal arrays

Reference:
Vazquez Alcocer Alan Roberto, Goos Peter, Schoen Eric D..- Constructing tw o-level designs by concatenation of strength-3 orthogonal arrays
Technometrics : a journal of statistics for the physical, chemical, and engineering sciences - ISSN 0040-1706 - 61:2(2019), p. 219-232
Full text (Publisher's DOI): https://doi.org/10.1080/00401706.2018.1473798
To cite this reference: https://hdl.handle.net/10067/1614010151162165141



Technometrics

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: http://amstat.tandfonline.com/loi/utch20

Constructing Two-Level Designs by Concatenation of Strength-3 Orthogonal Arrays

Alan R. Vazquez, Peter Goos & Eric D. Schoen

To cite this article: Alan R. Vazquez, Peter Goos & Eric D. Schoen (2018): Constructing Two-Level Designs by Concatenation of Strength-3 Orthogonal Arrays, Technometrics, DOI: 10.1080/00401706.2018.1473798

To link to this article: https://doi.org/10.1080/00401706.2018.1473798

Constructing Two-Level Designs by Concatenation of Strength-3 Orthogonal Arrays

Alan R. Vazquez[1], Peter Goos[1,2], and Eric D. Schoen[1,3]

[1]University of Antwerp, Belgium
[2]KU Leuven, Belgium
[3]TNO, Zeist, Netherlands

April 30, 2018

Abstract

Two-level orthogonal arrays of N runs, k factors and a strength of 3 provide suitable fractional factorial designs in situations where many of the main effects are expected to be active, as well as some two-factor interactions. If they consist of $N/2$ mirror image pairs, these designs are fold-over designs. They are called even and provide at most $N/2 - 1$ degrees of freedom to estimate interactions. For $k < N/3$ factors, there exist strength-3 designs that are not fold-over designs. They are called even-odd designs and they provide many more degrees of freedom to estimate interactions. For $N \leq 48$, attractive even-odd designs can be extracted from complete catalogs of strength-3 orthogonal arrays. However, for larger run sizes, no complete catalogs exist. In order to construct even-odd designs with $N > 48$, we develop an algorithm for an optimal concatenation of strength-3 designs involving $N/2$ runs. Our approach involves column permutations of one of the concatenated designs, as well as sign switches of the elements of one or more columns of that design. We illustrate the potential of the algorithm by generating two-level even-odd designs with 64 and 128 runs involving up to 33 factors, because this allows a comparison with benchmark designs from the literature. With a few exceptions, our even-odd designs outperform or are competitive with the benchmark designs in terms of the aliasing of two-factor interactions and in terms of the available degrees of freedom to estimate two-factor interactions.

Keywords: Even-odd design, generalized aberration, local search, second-order saturated, two-factor interaction, variable neighborhood search

1 Introduction

Two-level screening designs can be used as experimental plans to identify, from a list of potentially influential factors, those that are indeed influential; see Mee et al. (2017) for a recent review of these designs. Many two-level screening designs currently in use involve orthogonal arrays (OAs). Denoting the two levels for each factor by -1 and $+1$, these OAs have main effect contrast vectors that are level-balanced and orthogonal to each other.

This paper is motivated by two practical experiments involving many runs and factors that were conducted using an OA. The first one is an enzyme stability experiment conducted at TNO, Zeist, The Netherlands. In order to improve the stability of an enzyme in a watery solution at room temperature, 17 possible additives were considered. The experiment actually carried out had 64 runs. The second experiment involves a sensitivity analysis of a simulation model for a software process (Houston et al., 2001), and used a design with 30 factors and 64 runs. In this paper, we propose a method to construct high-quality designs for these two experiments. The method is suitable for constructing large two-level OAs involving many runs and factors in general. We revisit the enzyme stability experiment and the software process simulation experiment after introducing and evaluating our method.

OAs of strength t are such that all 2^t level combinations of any set of t factors occur equally often (Hedayat et al., 1999). Consequently, an OA of strength t has a run size that is a multiple of 2^t. A strength of 2 implies that main effect (ME) contrast vectors are orthogonal to each other but not to two-factor interaction (2FI) contrast vectors. This feature may make it hard to find out whether it is the main effect of one of the factors or the interaction of two other factors that causes a change in the responses. On the positive side, the run sizes of strength-2 OAs can be small. For example, the smallest 17-factor strength-2 OA involves 20 runs; a 20-run 17-factor design that minimizes the aliasing between the MEs and the 2FIs can be found in Sun et al. (2008). The smallest 30-factor strength-2 OA involves 32 runs. Textbooks such as Mee (2009) and Wu and Hamada (2009) provide suitable design options for this case.

OAs of strength 3 allow the main effects to be estimated independently from the 2FIs. They are attractive options whenever several 2FIs are suspected to be active, but they have larger run sizes. For 17 and 30 factors, the smallest strength-3 OAs have 40 and 64 runs, respectively. In general, for an N-run k-factor OA of strength 3, $N \geq 2k$. Butler (2004, 2007) showed that all strength-3 OAs for which $k \geq N/3$ must be even designs, which are also called fold-over designs (Cheng et al., 2008). In these designs, half of the runs are mirror images of the other half, in the sense that the signs of the factor levels are switched. A weakness of even designs is that they provide at most $N/2 - 1$ degrees of freedom for estimating 2FIs. For $k < N/3$ factors, there may be strength-3 designs that cannot be constructed by folding over. These designs, called even-odd designs, generally provide many more degrees of freedom for estimating 2FIs. Therefore, they are attractive to experimenters who want to estimate a substantial number of interactions along with the main effects. Due to the fact that their run size N should be larger than $3k$, even-odd designs must have at least 56 and 96 runs for experiments with 17 and 30 factors, respectively.

Complete catalogs exist for two-level strength-3 OAs with up to 48 runs (Schoen et al., 2010). Based on these catalogs, Schoen and Mee (2012) showed that, for run sizes of 32, 40 and 48, even-odd designs exist for up to 10, 10 and 14 factors, respectively. One way to obtain even-odd k-factor strength-3 designs for which the number of runs is larger than 48 is to concatenate two different strength-3 designs involving $N/2$ runs and $k - 1$ factors, which we call parent designs. Subsequently, a factor whose level equals -1 for the first $N/2$ runs and $+1$ for the last $N/2$ runs can be added to the concatenated design. Strength-3 designs constructed in this way have run sizes that are multiples of 16. This approach can thus be used to construct even-odd 64-run and 128-run designs based on existing strength-3 designs with 32 and 64 runs, respectively.

Several authors have constructed even-odd designs using variants of the above general approach. Li and Lin (2003), Li et al. (2003) and Cheng et al. (2008) concatenate two copies of a single parent design and subsequently switch the signs of all elements in one or more select columns of the second copy to improve the properties of the concatenated design. Their approach involves a complete enumeration of all possible selections of columns in which to switch the signs of the elements. Li and Lin (2016) suggested to also permute the columns of the second parent design before switching the signs in the selected columns. As a result, this approach also involves an enumeration of column permutations. In any

ACCEPTED MANUSCRIPT

case, the end product of the approaches of Li and Lin (2003), Li et al. (2003), Cheng et al. (2008) and Li and Lin (2016) is a concatenation of a parent design with another design that is isomorphic to that parent design (two OAs are said to be isomorphic if one array can be obtained from the other by permuting rows or columns, and switching the signs of the elements in one or more columns). There are two problems with these approaches. First, it may not be optimal to concatenate two isomorphic designs. Second, for strength-3 designs with run sizes larger than 48, the numbers of factors may be too large to allow for a complete enumeration of all possible sign switches and/or all column permutations. For example, for 17 factors, there are $131,072$ possible sets of columns in which to switch the signs and 3.55687×10^{14} possible column permutations.

The first contribution of this paper is to develop an efficient algorithmic procedure to construct even-odd designs by concatenating two strength-3 parent designs, which may or may not be isomorphic. At present, our algorithmic approach requires designs of the same run size and strength, but it can easily be adapted to concatenate designs of different strengths and run sizes.

The second contribution of this paper is to generate new two-level concatenated designs with 64 and 128 runs and up to 33 factors, for instance for the 17-factor enzyme stability experiment and the 30-factor software process simulation experiment. We compare the newly generated designs with the best designs from the literature. In addition to the benchmark designs of Li and Lin (2016), our comparisons involve the regular 64-run designs reported in Chen et al. (1993), the 17-factor 64-run design from Cheng et al. (2008), the regular 128-run designs reported in Block and Mee (2005) and Xu (2009), the 64- and 128-run designs constructed by Xu and Wong (2007) using quaternary linear codes, and the 64-run designs generated from projections of the folded-over 32-run Hadamard matrix given by the Paley construction. To our knowledge, our study of the projections from this folded-over 32-run Hadamard matrix is new to the literature. Most of our new designs outperform or are competitive with the best known designs in the literature in terms of the aliasing of two-factor interactions and in terms of the degrees of freedom they provide to estimate two-factor interactions.

The rest of this paper is organized as follows. Section 2 presents classification criteria for strength-3 designs. Section 3 describes our algorithmic approach for concatenating two strength-3 parent designs. Section 4 compares the new designs with 64 and 128 runs with the benchmark designs from the literature. We return to the motivating examples in Section 5, and conclude with a discussion and some suggestions for future research in Section 6.

2 Classification of strength-3 designs

Orthogonal two-level designs of strength 3 are most commonly evaluated in terms of their G-aberration and generalized resolution (Deng and Tang, 1999), their G_2-aberration (Tang and Deng, 1999), and the rank of the matrix consisting of the 2FI contrast vectors (Cheng et al., 2008). We briefly review all of these criteria.

All but the last of the criteria are based on the J_s-characteristics of s-factor interaction contrast vectors. When coding the two levels of each factor as -1 and $+1$, any s-factor interaction contrast vector involves the elements -1 or $+1$. Its J_s-characteristic is the absolute value of the sum of the vector's elements. For strength-3 OAs, any J_2- or J_3-characteristic is zero. Deng and Tang (1999) showed that the J_4-characteristics of N-run two-level strength-3 OAs necessarily equal $N-16q$, where q is a non-negative integer. A four-factor interaction contrast vector can be calculated as the product of two two-factor interaction contrast vectors. Therefore, whenever a J_4-characteristic of N occurs in an N-run design, this implies that three pairs of two-factor interactions are completely aliased. Whenever J_4-characteristics of zero occur in a design, this implies that certain pairs of two-factor interactions are not aliased at all. Intermediate J_4-characteristic values imply partially aliased two-factor interactions. The maximum J_4 value for a given strength-3 design determines its generalized resolution. More specifically, the generalized resolution equals $5 - \max(J_4)/N$ for a strength-3 design. So, a large J_4-characteristic implies a small generalized resolution. Ideally, the generalized resolution of a design is large.

The frequencies of the J_s-characteristics calculated for all s-factor interaction contrast vectors are generally collected in a vector F_s. For strength-3 designs, the entries of the F_4 vector are the frequencies of the J_4-characteristic values N, $N-16$, $N-32$, etc. The frequency of the zero value is usually omitted. The concatenated vector $(F_4, F_5, F_6, \ldots, F_k)$

is the confounding frequency vector (CFV) of a strength-3 k-factor design. Ideally, the leftmost elements of the CFV are small, because this means that there is little severe aliasing between the low-order interactions.

To determine the G-aberration of a k-factor design, all available designs are sorted according to the entries of the CFV $(F_4, F_5, F_6, \ldots, F_k)$, from left to right. The G-aberration of the design is its ranking after the sorting procedure, and a minimum G-aberration design has the best of the rankings. In this paper, we restrict our attention to the F_4 vector, because this vector quantifies the most serious aliasing in strength-3 designs, namely the aliasing among the 2FIs.

Like the G-aberration, the G_2-aberration is determined by sorting all available designs according to a vector. The vector used for the G_2-aberration is called the generalized word length pattern (GWLP), $(B_1, B_2, \ldots, B_k)$, where B_i is the sum of the squared J_i-characteristics divided by N^2. A minimum G_2-aberration design has the best of the rankings after sorting all available designs according to the entries of the GWLP from left to right. For strength-3 OAs, $B_1 = B_2 = B_3 = 0$, and $B_4 > 0$. The B_i values are called generalized word counts. The most important of these is B_4, because it quantifies the aliasing among the 2FIs. Ideally, it is small.

Finally, strength-3 OAs permit the estimation of the intercept and all the main effects simultaneously. The rank of the 2FI contrast matrix quantifies the number of degrees of freedom available for estimating two-factor interactions (Cheng et al., 2008). In the rest of this paper, we use the term 'degrees of freedom for two-factor interactions' to refer to this criterion.

3 Algorithmic construction of even-odd designs

In this section, we first describe the principles to concatenate two orthogonal arrays of strength 3. Next, we propose two interconnected algorithms to find an optimal permutation of the columns of one of the two parent designs, and the best subset of columns for sign switching. Our first algorithm is called the column change (CC) algorithm. It is a local search algorithm making small structured changes to one of the parent designs. The CC algorithm is embedded in a variable neighborhood search (VNS) algorithm that investigates increasingly diverse new versions of that parent design. The complete algorithm, called the CC/VNS algorithm, reduces the number of evaluations needed to test column permutations and subsets of columns in which to switch the signs. We conclude this section with an evaluation of the CC/VNS algorithm and with a study concerning the choice of parent designs.

3.1 Concatenation principles

Our CC/VNS algorithm concatenates two strength-3 orthogonal arrays, D_u and D_l, which both have $N/2$ runs and a given number of factors, say m. Adding the column $\mathbf{z} = [\mathbf{1}_{N/2}^T, -\mathbf{1}_{N/2}^T]^T$ to the concatenated m-factor design generated by our algorithm results in an N-run strength-3 design D with $k = m + 1$ factors. We call the added column the indicator factor, because it identifies the two original OAs. The m 2FIs involving this factor are all orthogonal to the 2FIs of the original factors. As a result, the B_4 value and the F_4 vector of a concatenated design are not affected by adding the indicator factor. Since a strength-3 orthogonal array with $N/2$ runs can accommodate at most $N/4$ factors, the concatenated design has N runs and involves at most $k = N/4 + 1$ factors, including the indicator factor. In this respect, our approach is similar to that of Cheng et al. (2008). We refer to D_u, D_l and D as the upper design, the lower design and the concatenated design, respectively. We call D_u and D_l parent designs.

The first step in the construction process is the selection of two strength-3 parent designs with $N/2$ runs. The parent designs may or may not be isomorphic. One of the parent designs serves as the upper design D_u, while the other becomes the lower design D_l. Which of the two parent designs is the upper and the lower design does not impact the quality of the resulting concatenated design. Next, we permute the columns of D_l and switch the signs of the elements in a certain number of columns of D_l, so as to improve the concatenated design in terms of a desired criterion. We refer to the lower design produced after switching the signs in a subset of its columns and applying a column permutation as a *plan* for D_l.

Whenever $m \geq N/6$, the $N/2$-run strength-3 parent designs for our procedure must be even. Since sign switches and column permutations in the parent designs do not change their even nature, m-factor concatenated designs are also even when $m \geq N/6$. A parent design provides at most $N/4 - 1$ degrees of freedom for 2FIs. Therefore, m-factor concatenated designs provide at most $2(N/4 - 1) = N/2 - 2$ degrees of freedom for 2FIs. Concatenated even designs become even-odd only after adding the indicator factor. The m 2FIs involving this factor can be estimated independently from all other 2FIs. Hence, the maximum number of degrees of freedom for 2FIs is $N/2 - 2 + m$ when $m \geq N/6$.

Concatenated designs with added indicator factors become second order saturated (SOS, Cheng et al., 2008) if the number of main effects, $m + 1$, plus the number of degrees of freedom for 2FIs equals $N - 1$. That can happen only if $N/2 - 2 + m = N - 1 - (m + 1)$, or $m = N/4$. Therefore, the concatenation of even designs with $m < N/4$ does not lead to an SOS design, while the concatenation of even designs with $m = N/4$ may or may not lead to an SOS design.

The total number of plans that can be obtained for a lower design D_l by permuting its columns and switching signs in sets of columns is $m! \times 2^m$. Evaluating all possible plans is computationally infeasible when $m \geq 10$. Rather than completely enumerating all plans for the lower design, our concatenation procedure uses the column change (CC) algorithm embedded within a variable neighborhood search (VNS) algorithm.

Our CC/VNS algorithm improves the concatenated design either in terms of the F_4 vector or in terms of the B_4 value. By optimizing the F_4 vector, the CC/VNS algorithm also automatically maximizes the generalized resolution of the concatenated design.

3.2 Column change algorithm

Algorithm 1 shows our CC algorithm, which is a local search algorithm (Michalewicz and Fogel, 2004) that evaluates changes to the current plan for the lower parent design in terms of the B_4 value or the F_4 vector. We developed fast update procedures for the B_4 value and the F_4 vector, so that the evaluation of the B_4 value and the F_4 vector can be done without computing the 2FI contrast matrix from scratch for every change applied by the algorithm. A detailed account of our update procedures is given in Section A of the supplementary materials. Algorithm 1 requires two m-factor designs with $N/2$ runs as inputs.

The algorithm starts by switching the signs in the leftmost column of the current plan for the lower design D_l and evaluates the resulting concatenated design. If the change does not yield an improvement, the algorithm starts evaluating swaps between the leftmost column and the columns to its right; see lines 11–19 in Algorithm 1. Two types of swaps are performed. The first swap involves the unmodified columns 1 and j, while the second swap involves the original column 1 and the sign-reversed column j. As soon as these modifications to the lower design result in an improvement of the concatenated design in terms of the B_4 value or the F_4 vector, the improved design replaces the original and the algorithm shifts its attention to the second column. First, it switches the signs in column 2 of the current plan for the lower design D_l and evaluates the resulting concatenated design. If the sign switch does not yield a better concatenated design, the algorithm evaluates swaps between column 2 and the columns to its right. This process is repeated for each of the columns in the current plan for the lower design D_l, and it ends with an evaluation of the concatenated design resulting from a sign switch of column m of the current plan for D_l.

Each time a sign switch of a certain column i or a swap of it with one of the (possibly sign-reversed) columns to its right results in an improved concatenated design, the algorithm continues its operations on this newly obtained improved design. The algorithm therefore uses a first-improvement optimization strategy. The algorithm makes several passes through all the columns and stops when no better plan can be found for the lower design D_l. The output of Algorithm 1 is an improved plan $D_l^\star$ of the original lower parent design D_l.

3.3 Variable neighborhood search algorithm

Variable neighborhood search or VNS is a metaheuristic introduced by Hansen and Mladenović (2001) as an improvement over local search algorithms for combinatorial optimization. A weakness of a local search algorithm is that it may get stuck in a locally optimal solution instead of a global optimum because it does not examine all possible changes to the existing solution. VNS attempts to overcome this weakness by systematically exploring more than one neighborhood structure. A neighborhood structure is defined by a type of change that can be made to a given solution s. Each allowable change is called a move.

Algorithm 1: Pseudocode of the CC algorithm.

Input: D_u and D_l

$D_l^\star \leftarrow D_l$
Set $i \leftarrow 1$
repeat
    for $i = 1, \ldots, m$ **do**
        Construct plan D_l' by switching signs in column i of $D_l^\star$.
        if *concatenated design* (D_u, D_l') *is better than* $(D_u, D_l^\star)$ **then**
            $D_l^\star \leftarrow D_l'$
        else
            Set $j \leftarrow i + 1$
            Set *no_improvement* $\leftarrow$ True
            while $j \leq m$ and *no_improvement* **do**
                Construct plan D_l^+ by swapping columns i and j of $D_l^\star$.
                Construct plan D_l^- by switching the signs in column j of $D_l^\star$ and swapping the resulting column with column i.
                Evaluate the concatenated designs (D_u, D_l^+) and (D_u, D_l^-).
                Set D_l' to be the best of the plans D_l^+ and D_l^-. If both concatenated designs are equally good, select at random.
                if *concatenated design* (D_u, D_l') *is better than* $(D_u, D_l^\star)$ **then**
                    $D_l^\star \leftarrow D_l'$
                    no_improvement $\leftarrow$ False
                $j \leftarrow j + 1$
until *no change in* $D_l^\star$

Output: Improved plan $D_l^\star$

All solutions s' that can be reached by one move are said to be in the neighborhood $N(s)$ of s. The rationale for using more than one neighborhood is that a solution which is a local optimum with respect to one neighborhood is not necessarily a local optimum with respect to another neighborhood. For this reason, escaping from a locally optimal solution

ACCEPTED MANUSCRIPT

can be done by changing the neighborhood structure. Unlike many other metaheuristics, VNS is simple to implement and requires few, and sometimes even no tuning parameters. Moreover, Hansen et al. (2008) showed that the VNS framework is very general and can be easily extended to integrate features from tabu search (Glover and Laguna, 1997), simulating annealing (Eglese, 1990), and other local search algorithms. VNS has been successfully applied to a wide variety of optimization problems such as vehicle routing (Kytöjoki et al., 2007), project scheduling (Fleszar and Hindi, 2004), automatic discovery of theorems (Caporossi and Hansen, 2004), graph coloring (Avanthay et al., 2003), and synthesis of radar polyphase codes (Mladenović et al., 2003).

On various occasions, VNS has also been used to construct experimental designs. For instance, Garroi et al. (2009) proposed a VNS algorithm to compute D-optimal run orders for response surface designs in the presence of serial correlation. More recently, Sartono et al. (2015) and Syafitri et al. (2015) used VNS to construct fractional factorial split-plot designs and optimal mixture designs in the presence of ingredient availability constraints, respectively.

Our CC/VNS algorithm performs systematic changes to the lower parent design D_l so as to minimize the F_4 vector or the B_4 value of the concatenated design. It involves two main components: (i) four neighborhood structures to create neighboring plans from the current best plan of D_l and (ii) the CC algorithm described in Section 3.2 to improve these neighboring plans. Two plans A and B of the lower design are said to be neighboring plans if $A \in N_i(B)$ or $B \in N_i(A)$ for a neighborhood structure N_i. Because of the two main components of our CC/VNS algorithm, it belongs to the general class of VNS algorithms in which a local search algorithm is used to improve the neighboring solutions created by the neighborhood structures (Hansen et al., 2008).

The four neighborhood structures used by our CC/VNS algorithm are listed in Table 1 and start by modifying one, two, two and three columns, respectively. As a result, the four neighborhoods explore increasingly diverse plans for the lower parent design D_l. Table 1 shows that the size of the first neighborhood structure increases linearly with the number of factors m of the parent designs. For this reason, the size of this neighborhood structure, N_1, is denoted by $O(m)$. The sizes of the second and third neighborhood structures, N_2

ACCEPTED MANUSCRIPT

Table 1: Neighborhood structures of the CC/VNS algorithm.

N_i	Size	Description
N_1	$O(m)$	Switch signs of any column
N_2	$O(m^2)$	Swap any two columns
N_3	$O(m^2)$	Switch signs of any two columns
N_4	$O(m^3)$	Choose any subset of three columns, move the first two columns one position to the right and move the third column to position 1

and N_3, increase according to a second-order polynomial in m, which is why these sizes are denoted by $O(m^2)$. The size of the last neighborhood, N_4, increases according to a cubic polynomial in m, which is denoted by $O(m^3)$.

The outline of our CC/VNS algorithm is shown in Algorithm 2. The input to the algorithm is an upper parent design D_u and a lower parent design D_l. The algorithm begins by generating a starting plan for D_l in three steps; see lines 1 and 2 of Algorithm 2. First, the signs of all elements in r randomly selected columns of D_l are switched, where r is a random integer between 0 and m. Second, the columns of the resulting plan are randomly permuted. Third, the resulting plan is optimized by the CC algorithm described in Section 3.2.

After the starting plan of the lower design D_l has been generated, the CC/VNS algorithm continues by exploring the first neighborhood structure (N_1) of the starting plan. To this end, it randomly selects a plan from the neighborhood and applies the CC algorithm to it, to attempt to find a better plan for the lower parent design. If a better plan is indeed found, the CC/VNS algorithm continues by exploring the first neighborhood structure of the newly obtained improved plan. If the CC algorithm does not produce a better plan, a second plan is selected from the first neighborhood structure of the starting plan and an attempt is made to improve it using the CC algorithm. The exploration of the first neighborhood structure of a given plan continues until all plans it contains have been optimized by means of the CC algorithm. If this does not yield any better plan than the current

Algorithm 2: Pseudocode of the CC/VNS algorithm.

Input: D_u and D_l

$R_l \leftarrow$ Random plan for D_l

Generate starting plan D_l^* using the CC algorithm and D_u and R_l as input.

Set $i \leftarrow 1$

while $i \leq 4$ **do**

  Set *improvement* $\leftarrow$ False

  repeat

    Randomly select a plan S_l from $N_i(D_l^*)$.

    Generate an improved plan D_l' using the CC algorithm and D_u and S_l as input.

    if *concatenated design* (D_u, D_l') *is better than* (D_u, D_l^*) **then**

      $D_l^* \leftarrow D_l'$

      improvement $\leftarrow$ True

      $i \leftarrow 0$

  until *no unexplored plans left in* $N_i(D_l^*)$ or *improvement*

  $i \leftarrow i + 1$

Output: concatenated design (D_u, D_l^*)

best one, the algorithm starts exploring the second neighborhood structure (N_2), in the same fashion. As soon as the exploration of the second neighborhood structure results in an improved plan, the CC/VNS algorithm returns to the first neighborhood structure and explores that first neighborhood structure of the improved plan. If the exploration of the second neighborhood structure does not produce any improved plan, the third neighborhood structure (N_3) is explored, and, if that does not lead to any improved plan, the fourth neighborhood structure (N_4) is explored. The process is repeated until no further improvement can be reached. In the course of the optimization, each neighborhood structure involves high-quality neighbors of the current best plan for the lower parent design, which has a positive impact on the performance of our CC/VNS algorithm.

Finally, to increase the likelihood of finding a globally optimal plan for the lower parent design, the CC/VNS algorithm is repeated a number of times, each time starting from a randomly generated plan for the lower parent design. This multi-start procedure in the algorithmic construction is common to virtually all design construction algorithms in the literature. Eventually, the overall best plan found for the lower parent design over all iterations is reported.

A Matlab implementation of the CC/VNS algorithm is included in the supplementary materials of this paper. Matlab allows parallel computations to decrease the calculation time. For specific parent designs, we present a comprehensive evaluation of the neighborhood structures of the CC/VNS algorithm and the computing times in Supplementary Section B. A key result from our evaluation is that each of the four neighborhoods of the CC/VNS algorithm contributes significantly to the quality of the concatenated designs generated.

3.4 Performance of the CC/VNS algorithm

In this section, we first evaluate the performance of our CC/VNS algorithm to generate the best 32- and 64-run concatenated designs with up to 11 factors from strength-3 regular designs with 16 and 32 runs. To this end, we compare our concatenated designs to those obtained by Li and Lin (2016). Next, we assess the performance of our algorithm to generate the best 128-run concatenated designs with up to 30 factors from regular and nonregular designs of strength 3 with 64 runs.

3.4.1 Comparison with a benchmark approach

We first evaluate the potential of our CC/VNS algorithm by testing whether it is able to match or even improve the results of Li and Lin (2016, LL16), using the strength-3 parent designs they used. LL16 constructed 32-run designs with up to 9 factors and 64-run designs with up to 12 factors by concatenating regular resolution IV 2^{m-p} designs with 16 runs and up to 8 factors and regular resolution IV 2^{m-p} designs with 32 runs and up to 11 factors, respectively. They used the same design as upper parent design D_u and as lower parent design D_l and searched for a plan for D_l that sequentially minimizes the CFV of the concatenated design. For this reason, we now focus on the minimization of the F_4

vector, which is the most important component of the CFV for strength-3 designs. LL16 showed that, when the same regular 2^{m-p} design is used as upper and lower parent design, the number of computations required to evaluate all possible sign switches of columns can be reduced from 2^m to 2^p. For parent designs with up to 9 factors, they evaluated all $m! \times 2^p$ possible plans for D_l. Evaluating all $m!$ column permutations for parent designs with more than nine factors was computationally infeasible. For this reason, for ten factors or more, they used a large number of randomly chosen permutations instead of all possible permutations. They ran their enumeration program for 10- and 11-factor parent designs for 168 hours and reported the best results found. They did not report the computing times for the complete enumeration of concatenated designs based on parent designs with up to 9 factors.

The results of 1,000 iterations of the CC/VNS algorithm applied to the cases of LL16 are shown in Table 2. The computing times for the 6-, 7-, and 8-factor designs were less than a second per iteration on a standard CPU (Intel(R) Core(TM) i7 processor, 2.8 Ghz, 8 GB). For 9 factors or more, the computing times ranged from 1 to 6 seconds per iteration.

The first column of the table shows the run size N of the concatenated design. The second column shows the parent designs used for the concatenation in the form m-$p.z$, where m is the number of factors, p is the number of generators of the design and z is the ranking of the design according to the aberration criterion (Chen et al., 1993). The third column presents the frequencies $F_4^{\max}$ of the largest values of the J_4-characteristics of the concatenated designs. The largest J_4-characteristic equals 16 for the best 32-run designs we found and 32 for the best 64-run designs, except for the designs constructed from the parent designs 7-2.1 and 7-2.2. These parent designs resulted in 64-run designs in which all J_4-characteristics equal zero. As a result, all designs we found have a generalized resolution of at least 4.5. None of the 32-run designs we obtained have J_4-characteristics of 32, so that there is no complete aliasing between 2FIs. For all 64-run designs we obtained, $F_4(64, 48, 16) = (0, 0, 0)$, meaning that J_4-characteristics of 64, 48 and 16 do not occur. So, there is also no complete aliasing between 2FIs in the 64-run designs. The F_4 vectors of the best designs we obtained coincide with those found by LL16, except for the 11-factor 64-run design based on parent design 11-6.2. The design produced by our CC/VNS

ACCEPTED MANUSCRIPT

algorithm outperforms the benchmark design of LL16, for which $F_4(32) = 46$. For our design, $F_4(32) = 44$. So, in our design, there is less aliasing between the 2FIs.

The fourth column of Table 2 shows the percentage of iterations during which the CC/VNS algorithm was able to find the best F_4 vector. For all but three of the cases in Table 2, this percentage equals 100. In other words, the CC/VNS algorithm generally obtained the best possible concatenated design at each iteration. For the parent designs 7-2.1, 10-5.4 and 11-6.2, the percentages were 88.1, 96.8 and 65.9, respectively. These large percentages imply that it is almost certain that 10 iterations of the CC/VNS algorithm suffice to find the best concatenated design. As a matter of fact, even for the worst case in Table 2, the probability to obtain the best design at least once in 10 iterations is $1-(1-0.659)^{10} \approx 1$. Remarkably, in each iteration where the CC/VNS algorithm failed to find the best design with $F_4(32) = 44$ for this case, it produced LL16's design with $F_4(32) = 46$. This shows that, even when the algorithm fails to find the best design, it produces a high-quality alternative.

Column 5 of Table 2 shows the number of plans for the lower parent design D_l explored by the algorithm, relative to $m! \times 2^p$ and expressed as a percentage. In all but one case, less than 40% of the total number of plans are evaluated by the CC/VNS algorithm. The exception is parent design 6-2.1, for which the CC/VNS algorithm evaluated 34% more plans than a complete enumeration would ($6! \times 2^2 = 2880$). However, in each iteration of the CC/VNS algorithm, the CC algorithm provided a starting plan with an optimal F_4 vector. Constructing this plan only required 2.35% of the total number of evaluations. The additional computations are due to the CC/VNS algorithm's visits to the four neighborhood structures, to confirm the excellent quality of the starting solution produced by the CC algorithm.

Our comparison with the benchmark approach and the comprehensive evaluation of the CC/VNS algorithm in Section B of the supplementary materials allows us to conclude that our CC/VNS algorithm creates high-quality concatenated designs using a considerably smaller computing effort than that needed by LL16.

Table 2: Results produced by the CC/VNS algorithm for 32-run and 64-run concatenated designs constructed from regular resolution IV designs. The labels of the parent designs are those used by Chen et al. (1993). $F_4^{\max}$ represents the frequency of the largest values of the J_4-characteristics. The number of plans evaluated to find the optimal F_4 vector is expressed as a percentage of the total number of possible plans averaged over 1,000 iterations.

N	Parent	$F_4^{\max}$	Percentage of iterations	Percentage of plans
32	6-2.1	4	100	134.104
	7-3.1	12	100	18.953
	8-4.1	24	100	2.132
64	7-2.1	0	88.1	24.580
	7-2.2	0	100	27.421
	7-2.3	4	100	39.416
	8-3.1	4	100	4.643
	8-3.2	6	100	4.765
	8-3.3	8	100	4.649
	8-3.4	12	100	4.492
	9-4.1	8	100	0.448
	9-4.2	12	100	0.441
	9-4.3	12	100	0.464
	9-4.4	16	100	0.444
	9-4.5	24	100	0.419
	10-5.1	16	100	0.036
	10-5.2	24	100	0.036
	10-5.3	26	100	0.037
	10-5.4	30	96.8	0.044
	11-6.1	42	100	0.002
	11-6.2	44	65.9	0.003

3.4.2 Performance for 128-run designs

We also tested the potential of the CC/VNS algorithm by constructing 128-run designs involving 10, 15, 20, 25 and 30 factors from 64-run parents. We obtained suitable 10-, 15-, 20-, 25- and 30-factor parent designs from the complete collection of regular 64-run resolution IV 2^{m-p} designs (Chen et al., 1993), the collection of nonregular designs based on quaternary linear codes (QLC) found by Xu and Wong (2007), and the 64-run designs generated from projections of the folded-over 32-run Hadamard matrix given by the Paley construction (Sloane, 1999).

Supplementary Section B shows our detailed results for 100 iterations of the CC/VNS procedure, for the case where we optimized the B_4 value of the concatenated designs as well as for the case where we optimized the F_4 vector. When optimizing the B_4 value of designs with up to 20 factors, the best design was found in 65% or more of the iterations of the CC/VNS algorithm. The cases with 25 and 30 factors were clearly more challenging, as the success rate dropped to values as low as 6% and 13% for these cases. When optimizing the F_4 vector, the success rates are even lower than that, due to the fact that optimizing the F_4 vector is harder than optimizing the B_4 value. The small probability of identifying the best concatenated design is not a major problem, because, whenever it fails to find the best design, the CC/VNS algorithm still produces a high-quality concatenated design with the same generalized resolution as the best one and with a B_4 value and an F_4 vector that are only slightly worse than those of the best concatenated design. From our results, we concluded that 10 iterations will generally suffice to find a high-quality 128-run design. In the event there are no more than 20 factors, that design will most likely be the best.

For all cases studied here, only a very small proportion (generally much smaller than 1%) of all possible plans for the lower parent design D_l are evaluated by the CC/VNS algorithm when constructing the concatenated design. The computing times for optimizing the B_4 value ranged from 1 to 709 seconds for one iteration of the algorithm. For optimizing the F_4 vector, the computing times varied from 2 to 904 seconds per iteration, for up to 20 factors. For 25 and 30 factors, the computing times went up to 3.3 hours.

3.5 Choice of parent designs

To investigate how the quality of concatenated designs depends on the choice of the parent designs, we constructed 64- and 128-run concatenated designs with parent designs that differ in G- or G_2-aberration. The parent designs we used in this study to construct 64-run designs were selected from the complete catalog of 32-run strength-3 designs (Schoen et al., 2010) with 8, 10, 12, 14 and 16 factors, while the parent designs we used to construct 128-run designs were selected from the complete catalog of regular 64-run resolution IV 2^{m-p} designs (Chen et al., 1993) with 16, 18 and 20 factors. In this section, we discuss the results for concatenated designs with 10 factors and 64 runs, as well as those for concatenated designs with 16 factors and 128 runs. Results for all other cases follow the same pattern and allow for the same conclusions.

The complete catalog of 32-run designs with 10 factors includes 32 designs. Their B_4 values range from 10 to 18 and the frequencies of the J_4-characteristics of 32 range from 1 to 18. The complete catalog of 64-run regular designs with 16 factors includes 48 designs. Their B_4 values and, equivalently, the frequencies of the J_4-characteristics of 64 range from 43 to 105. From each of these two catalogs, we selected a set of five parent designs to construct F_4-optimized concatenated designs and a set of five parent designs for B_4-optimized concatenated designs. For B_4-optimized concatenated designs, the selected parent designs were the best and worst designs according to the G_2-aberration criterion and those corresponding to the first, second and third quartiles in that ranking. For minimizing the F_4 value, we selected the designs in a similar fashion based on the G-aberration criterion. We concatenated all 15 possible pairs of the five selected parent designs, including pairs of the same designs.

Table 3 shows the results for 100 iterations of the CC/VNS algorithm when the objective is to minimize the B_4 value. The table shows that the smallest B_4 values in the 64- and 128-run concatenated designs result from concatenating two copies of the minimum G_2-aberration designs. Concatenating minimum G_2-aberration designs with any of the four other designs results in larger B_4 values for the concatenated designs. Table 3 also shows that using bad 32- and 64-run designs in terms of the G_2-aberration criterion, such as the worst design (W) and the design corresponding with the third quartile (Q_3) of the G_2-

ACCEPTED MANUSCRIPT

aberration ranking, generally results in 64- and 128-run designs with the largest B_4 values. It is interesting to mention though that the 64-run concatenated design constructed from the 32-run parent designs B and Q_2 has a better B_4 value than the concatenated design constructed from the parent designs B and Q_1, even though design Q_2 has a worse G_2-aberration than design Q_1. This implies that the G_2-aberration ranking of the parent designs does not necessarily agree with the ranking of the resulting concatenated designs in terms of the B_4 value.

Table 3: B_4 values for the B_4-optimized concatenated designs with 64 and 128 runs. The symbols B, Q_1, Q_2, Q_3 and W correspond to the best ranked design, the designs corresponding to the first, second and third quartiles and the worst ranked design, respectively, in terms of G_2-aberration.

(a) 10 factors and 64 runs

Parents	B	Q_1	Q_2	Q_3	W
B	4				
Q_1	5.69	6			
Q_2	5.5	6.19	6		
Q_3	5.63	6.31	6.38	6.38	
W	6	6.69	7	7.13	7.5

(b) 16 factors and 128 runs

Parents	B	Q_1	Q_2	Q_3	W
B	17				
Q_1	21.75	20			
Q_2	22	25.25	25		
Q_3	23	26.25	26	27	
W	32.5	33.75	36.5	37.5	45

Table 4 shows the results for 100 iterations of the CC/VNS algorithm when the objective is to minimize the F_4 vector. The conclusions that can be drawn from that table are similar to those for the B_4-optimized concatenation. That is, better parent designs in terms of the F_4 vector lead to better F_4 vectors for the concatenated designs. For instance, the table shows that concatenating two copies of the best parent design with 32 runs leads to a 64-run concatenated design without any J_4-characteristic of 64 and a generalized resolution as large as 4.75. For the 128-run design case, concatenating two copies of the best 64-run regular design leads to a much lower frequency of the J_4-characteristics of 64 than concatenating any other pair of selected parents.

The specific cases discussed here as well as our more comprehensive study of the relation

Table 4: Frequencies of J_4-characteristics of 32 and 64 for the F_4-optimized concatenated designs with 64 and 128 runs, respectively. The symbols B, Q_1, Q_2, Q_3 and W correspond to the best ranked design, the designs corresponding to the first, second and third quartiles and the worst ranked design, respectively, in terms of G-aberration.

(a) 10 factors and 64 runs

Parents	B	Q_1	Q_2	Q_3	W
B	0				
Q_1	2	4			
Q_2	4	6	8		
Q_3	7	9	12	16	
W	14	15	19	24	30

(b) 16 factors and 128 runs

Parents	B	Q_1	Q_2	Q_3	W
B	72				
Q_1	89	96			
Q_2	90	105	108		
Q_3	94	109	110	116	
W	134	147	150	156	196

between the quality of concatenated designs and the choice of the parent designs permit the following conclusion: Concatenating the best parent designs in terms of the G- and G_2-aberration generally leads to the best concatenated designs in terms of the F_4 vector and B_4 value, respectively. For this reason, when constructing concatenated designs using our CC/VNS algorithm, we recommend to use the best parent designs available in terms of the desired optimization criterion.

4 Results

Encouraged by the test results, we used the CC/VNS algorithm to generate two-level even-odd designs for run sizes 64 and 128, based on the best parent designs available with 32 and 64 runs, respectively. A detailed description of the parent designs is included in Supplementary Section C. Tables showing the sign switches and permutations in the lower parent design along with the full F_4 vectors of the concatenated designs are presented in Supplementary Section D. In the present section, we discuss the most important features of the concatenated designs and compare them with benchmark designs from the literature.

For each combination of number of runs and number of factors, we considered several

ACCEPTED MANUSCRIPT

pairs of attractive parent designs. We concatenated each pair of attractive parent designs to investigate which pair gives rise to the best concatenated design. For each given pair of parent designs, we used 40 iterations of the CC/VNS algorithm when the objective was to minimize the B_4 value and 10 iterations when the objective was to sequentially minimize the F_4 vector. We used a larger number of iterations when minimizing the B_4 value because calculating the B_4 value is computationally less demanding than calculating the F_4 vector, especially when the number of factors exceeds 20 (see Supplementary Section A). After running the CC/VNS algorithm, we constructed our final k-factor design by adding the indicator factor to the resulting m-factor concatenated design. Section D of the supplementary materials shows that nine of our best 66 concatenated designs are constructed from different, non-isomorphic parent designs. In all other cases, the upper and lower parent designs were isomorphic.

4.1 64 runs

Table 5 shows the B_4 value, the generalized resolution (GR), the frequency of the largest J_4-characteristic ($F_4^{\max}$) and the degrees of freedom for two-factor interactions (df) of 48 64-run designs involving 9–17 factors. It should be pointed out that comparing the $F_4^{\max}$ values of two designs only make sense if they have the same GR value and thus the same maximum J_4-characteristic. In that case, the design with the smaller $F_4^{\max}$ value has the better F_4 vector of the two.

The tabulated results are for the best concatenated designs the CC/VNS algorithm produced in terms of the F_4 vector (denoted by CC/F4) and in terms of the B_4 value (denoted by CC/B4). The parent designs we used as inputs were the best designs from the enumeration of Schoen et al. (2010); see Supplementary Section C. The table also includes the following benchmark designs:

- The regular designs listed by Chen et al. (1993). The designs included in the table have the smallest possible B_4 value and, subject to this, the largest number of degrees of freedom for 2FIs. Therefore, they are either minimum aberration designs (denoted by MA) or alternative designs with a larger number of degrees of freedom for 2FIs than the MA design (denoted by EST).

- The QLC designs of Xu and Wong (2007). We denote the QLC design with the best B_4 value as QLC/B4 and the design with the best F_4 vector as QLC/F4. We use the label QLC whenever there is a single design that is best in terms of the B_4 value and in terms of the F_4 vector.
- The 17-factor design of Cheng et al. (2008), denoted by CMY.
- The best projections of the folded-over 32-run Paley matrix.

The third column of Table 5 shows whether the designs are admissible (Sun et al., 1997), meaning that they are not dominated by another design when considering the B_4 value, the generalized resolution, the F_4 vector and the degrees of freedom for 2FIs. A design that is dominated by another design is called inadmissible. For example, the 9-factor MA design has the same B_4 value as the CC/B4 design, but its GR value and number of degrees of freedom for 2FIs are lower (note that the $F_4^{\max}$ value for the MA design should not be compared to that of the CC/B4 design, because the GR value and thus also the maximum J_4-characteristic are different for the two designs). Therefore, the 9-factor MA design is inadmissible.

Overall, 31 of the 48 64-run designs under comparison are admissible. Table 5 shows that all but two of the designs produced by the CC/VNS algorithm are admissible. The 10-factor CC/F4 design is inadmissible because it is dominated by the CC/B4 design, while the 14-factor CC/B4 design is inadmissible because it is dominated by the QLC design. There are seven inadmissible QLC designs, all of which are dominated by the CC/B4 designs. Three regular MA designs are inadmissible as they are dominated by the CC/B4 design, and one regular EST design is inadmissible because it is dominated by the QLC design. The 9-, 10- and 11-factor designs based on the folded-over Paley matrix are dominated by the CC/F4 designs. Finally, the 17-factor CMY design is dominated by the CC/B4 design.

Using the folded-over 32-run Paley matrix always results in the largest generalized resolution of 4.75, but, for 9, 10 and 11 factors, the CC/B4 and/or the CC/F4 design have the same generalized resolution, a better F_4 vector and a larger number of degrees of freedom for 2FIs. For larger numbers of factors, the CC/VNS algorithm does not produce designs with that large a generalized resolution, because the CC/VNS algorithm involves

ACCEPTED MANUSCRIPT

Table 5: 64-run designs involving 9–17 factors. B_4 values rounded to nearest integer. CC/B4: design by CC/VNS under B_4 optimization; CC/F4: design by CC/VNS under F_4 optimization; MA: regular minimum aberration design; EST: regular design with same B_4 value as MA design, but larger number of degrees of freedom for 2FIs; QLC: design based on quaternary linear code with best B_4 value and F_4 vector; QLC/B4: QLC design with best B_4 value; QLC/F4: QLC design with best F_4 vector; CMY: design from Cheng et al. (2008); P: projection of folded-over 32-run Paley matrix; ^{a}: design permits estimation of all 2FIs; ^{s}: SOS design.

k	Construction	Admissible	B_4	GR	$F_4^{\max}$	df	k	Construction	Admissible	B_4	GR	$F_4^{\max}$	df
9	CC/B4^a	Yes	1	4.75	16	36	14	CC/B4	No	22	4.5	88	43
	MA	No	1	4	1	33		EST	No	22	4	22	45
	QLC^a	No	1	4.5	4	36		QLC^s	Yes	14	4.5	56	49
	P	No	4	4.75	58	31		P	Yes	33	4.75	526	31
	CC/F4^a	Yes	1	4.75	16	36		CC/F4	Yes	24	4.5	24	43
10	CC/B4^a	Yes	2	4.75	32	45	15	CC/B4	Yes	33	4.25	8	44
	MA	No	2	4	2	39		MA	Yes	30	4	30	43
	QLC	No	2	4.5	8	39		QLC	No	33	4	21	43
	P	No	6	4.75	96	31		P	Yes	45	4.75	726	31
	CC/F4	No	2	4.75	32	44		CC/F4	Yes	35	4.5	38	44
11	CC/B4	Yes	4	4.5	16	48	16	CC/B4	Yes	45	4	9	45
	MA	No	4	4	4	44		MA	Yes	43	4	43	43
	QLC	No	4	4.5	16	47		QLC/B4	No	47	4	31	43
	P	No	10	4.75	160	30		QLC/F4	No	60	4	28	31
	CC/F4	Yes	7	4.75	108	40		P	Yes	61	4.75	978	31
12	CC/B4	Yes	10	4.5	21	41		CC/F4	Yes	49	4.5	57	45
	MA	Yes	6	4	6	50	17	CC/B4^s	Yes	60	4	12	46
	QLC	Yes	6	4.5	24	48		CMY^s	No	60	4	28	46
	P	Yes	16	4.75	252	31		MA	Yes	59	4	59	43
	CC/F4	Yes	11	4.5	5	41		QLC/B4	Yes	59	4	59	43
13	CC/B4	Yes	15	4.5	36	42		QLC/F4	No	64	4	40	43
	EST^s	Yes	14	4	14	50		P	Yes	80	4.75	1286	31
	QLC	Yes	10	4.5	40	48		CC/F4^s	Yes	65	4.5	83	46
	P	Yes	23	4.75	370	31							
	CC/F4	Yes	16	4.5	10	42							

strength-3 parent designs and the folded-over Paley matrix is essentially a concatenation of two strength-2 designs.

For 9 and 10 factors, the B_4 values we obtained are the minimum values possible (Xu, 2005). For 12–14 and 17 factors, the B_4 values of the designs produced by the CC/VNS algorithm are larger than those of the QLC designs and/or those of the regular MA or EST designs. This might imply that we did not use the best possible input designs for the CC/VNS algorithm. However, it is impossible to split the 64-run QLC designs with 12–14 and 17 factors into two strength-3 designs, so that these designs cannot be constructed by concatenating two strength-3 32-run designs. Likewise, it is not possible to split the regular resolution IV designs with 12, 13, 15, 16, and 17 factors into two strength-3 32-run designs. Therefore, the fact that the CC/VNS algorithm did not produce designs with lower B_4 values should not be viewed as a weakness of the algorithm, but as a consequence of our focus on concatenating strength-3 designs.

The column labeled df in Table 5 shows that the CC/VNS algorithm produces designs with 9–11 and 15–17 factors that provide at least as many degrees of freedom for 2FIs as the best benchmark designs. For 12–14 factors, the numbers of degrees of freedom for 2FIs of the CC/B4 and CC/F4 designs are smaller than those of the regular designs and the QLC designs. However, the numbers of degrees of freedom for 2FIs of the concatenated designs with 12–14 factors are the maximum numbers obtainable by concatenating even parent designs; see Section 3.1. As there are no even-odd strength-3 parent designs with 32 runs and more than 10 factors, it is not possible to construct concatenated designs with larger numbers of degrees of freedom for 2FIs for these cases when using strength-3 designs as parent designs.

Table 5 also identifies designs with which all 2FIs are estimable as well as SOS designs. The CC/VNS algorithm produced 9- and 10-factor designs which allow all 2FIs to be estimated. Our 17-factor designs, the 17-factor design of Cheng et al. (2008), the 13-factor regular resolution IV design, and the 14-factor QLC design of Xu and Wong (2007) are SOS designs. The 13- and 14-factor SOS designs cannot be constructed by concatenating two strength-3 designs.

ACCEPTED MANUSCRIPT

4.2 128-run designs

The parent designs we used for the CC/B4 designs with 128 runs were the 64-run regular minimum aberration designs (Chen et al., 1993), the 64-run QLC designs (Xu and Wong, 2007), and our own 64-run designs that minimize the B_4 value. For the 128-run concatenated designs that optimize the F_4 vector, the parent designs we used were the best projections of the folded-over 32-run Paley matrix and the 64-run concatenated designs produced by the CC/VNS algorithm. A detailed report of the best 128-run designs we obtained and their parent designs is given in Sections C and D of the supplementary materials.

4.2.1 10–15 factors

It is known that strength-4 128-run designs exist with up to 15 factors; see Hedayat et al. (1999) for the construction of the 15-factor design. These designs necessarily consist of two concatenated strength-3 64-run designs augmented with an indicator factor. Therefore, provided the right strength-3 64-run parent designs are used as input, the CC/VNS algorithm should be able to construct strength-4 128-run designs. This proved to be the case for our CC/B4 designs with 10–15 factors and our CC/F4 designs with 10 and 11 factors constructed using QLC and CC/B4 parent designs with 64 runs. For 12–15 factors, our CC/F4 designs only have a resolution of 4.75. The parents of these strength-3 designs are the best projections of the folded-over 32-run Paley matrix and our CC/F4 designs with 64 runs.

Supplementary Section E provides a comprehensive discussion of the strength-4 designs. It compares the 10- and 11-factor regular resolution V designs, the QLC designs involving 10–15 factors and the minimum G-aberration designs we identified based on the complete catalog of 128-run strength-4 OAs produced by Schoen et al. (2010). To the best of our knowledge, we are the first to identify the minimum G-aberration 128-run designs of strength 4.

4.2.2 16–33 factors

Table 6, which has the same format as Table 5, shows the main results for designs with 16–33 factors. The table includes our own concatenated designs as well as the following benchmark designs:

- The regular designs listed by Xu et al. (2009). The designs we used as benchmarks have the smallest possible B_4 value and, subject to this, the largest number of degrees of freedom for 2FIs. Therefore, they are either MA or EST designs.

- The QLC designs of Xu and Wong (2007).

Table 6 includes 84 designs, 19 of which are inadmissible. Seven of the 36 designs produced by the CC/VNS algorithm are inadmissible. The 16-factor CC/B4 design is dominated by the QLC/F4 design. The CC/B4 designs involving 17, 22–24, 26 and 27 factors are dominated by the corresponding QLC/B4 designs. The regular (MA or EST) designs for 16, 17, 20, 21 and 22 factors are dominated by the corresponding QLC/B4 designs and the 26-, 27- and 28-factor regular designs are dominated by the corresponding QLC designs. Four QLC designs are inadmissible: the QLC/B4 designs for 29 and 30 factors are dominated by the MA designs, while the QLC/F4 designs for 21 and 30 factors are dominated by the CC/B4 designs. The table further suggests that the QLC/F4 design with 30 factors is also dominated by the CC/F4 design, but this is due to the rounding of the B_4 value to the nearest integer.

All CC/F4 designs outperform all benchmark designs as well as the CC/B4 designs in terms of generalized resolution. Therefore, all CC/F4 designs are admissible. Except for the 12-factor case, the best parent designs for the CC/F4 designs are the 64-run designs generated from projections of the folded-over 32-run Paley matrix. So, all but one of the CC/F4 designs are constructed from a Paley-based design. If we denote the number of factors from that parent design by m, then we can verify in the column labeled df in Table 6 that the degrees of freedom for 2FIs is $2 \times 31 + m$ for each CC/F4 design.

The fact that our CC/B4 designs involving 16, 17, 22–24, 26 and 27 factors are inadmissible is not due to a bad choice of strength-3 parent designs. To reach this conclusion, we verified that none of the 128-run QLC designs can be split in two 64-run strength-3

Table 6: 128-run designs involving 16-33 factors. B_4 values rounded to nearest integer. CC/B4: design by CC/VNS under B_4 optimization; CC/F4: design by CC/VNS under F_4 optimization; MA: regular minimum aberration design; EST: regular design with same B_4 value as MA design, but larger number of degrees of freedom for 2FIs; QLC: design based on quaternary linear code with best B_4 value and F_4 vector; QLC/B4: QLC design with best B_4 value; QLC/F4: QLC design with best F_4 vector; ^{s}: SOS design.

k	Construction	Admissible	B_4	GR	$F_4^{\max}$	df	k	Construction	Admissible	B_4	GR	$F_4^{\max}$	df	k	Construction	Admissible	B_4	GR	$F_4^{\max}$	df
16	CC/B4	No	12	4	2	94	22	CC/B4	No	90	4	46	83	29	CC/B4	Yes	330	4	38	90
	MA	No	10	4	10	90		MA	No	65	4	65	102		MA^s	Yes	266	4	266	98
	QLC/B4	Yes	10	4	2	90		QLC/B4	Yes	56	4	42	105		QLC/B4	No	290	4	290	87
	QLC/F4	Yes	11	4.5	32	98		QLC/F4	Yes	66	4	28	96		QLC/F4	Yes	315	4	123	89
	CC/F4	Yes	19	4.75	131	77		CC/F4	Yes	97	4.75	753	83		CC/F4	Yes	343	4.75	2800	90
17	CC/B4	No	17	4	6	99	23	CC/B4	No	110	4	110	83	30	CC/B4	Yes	386	4	70	91
	EST	No	15	4	15	95		MA	Yes	83	4	83	102		MA	Yes	335	4	335	87
	QLC/B4	Yes	15	4	3	102		QLC	Yes	83	4	36	97		QLC/B4	No	336	4	336	87
	QLC/F4	Yes	16	4.5	48	99		CC/F4	Yes	119	4.75	942	84		QLC/F4	No	396	4	145	89
	CC/F4	Yes	28	4.75	189	78	24	CC/B4	No	136	4	76	85		CC/F4	Yes	396	4.75	3280	91
18	CC/B4	Yes	23	4	11	99		EST	Yes	102	4	102	102	31	CC/B4	Yes	447	4	43	92
	EST	Yes	20	4	20	97		QLC	Yes	101	4	45	97		MA	Yes	391	4	391	87
	QLC/B4	Yes	20	4	4	93		CC/F4	Yes	145	4.75	1150	85		QLC/B4	Yes	391	4	391	87
	QLC/F4	Yes	24	4.5	64	92	25	CC/B4	Yes	165	4	51	86		QLC/F4	Yes	417	4	161	91
	CC/F4	Yes	37	4.75	263	79		MA^s	Yes	124	4	124	102		CC/F4	Yes	458	4.75	3796	92
19	CC/B4	Yes	30	4	14	100		QLC	Yes	123	4	55	98	32	CC/B4	Yes	517	4	60	93
	MA	Yes	27	4	27	99		CC/F4	Yes	175	4.75	1405	86		MA	Yes	452	4	452	87
	QLC/B4	Yes	25	4	15	103	26	CC/B4	No	198	4	109	87		QLC/B4	Yes	452	4	452	87
	QLC/F4	Yes	32	4.5	96	98		MA	No	152	4	152	98		QLC/F4	Yes	480	4	192	91
	CC/F4	Yes	48	4.75	355	80		QLC	Yes	146	4	66	98		CC/F4	Yes	528	4.75	4372	93
20	CC/B4	Yes	40	4	20	105		CC/F4	Yes	209	4.75	1695	87	33	CC/B4^s	Yes	592	4	52	94
	MA	No	36	4	36	99	27	CC/B4	No	237	4	237	88		MA	Yes	518	4	518	87
	QLC/B4	Yes	32	4	18	103		MA	No	180	4	180	98		QLC/B4	Yes	518	4	518	87
	QLC/F4	Yes	39	4	15	95		QLC	Yes	174	4	78	99		QLC/F4	Yes	540	4	220	93
	CC/F4	Yes	61	4.75	457	81		CC/F4	Yes	248	4.75	2018	88		CC/F4^s	Yes	606	4.75	5044	94
21	CC/B4^s	Yes	52	4	20	106	28	CC/B4	Yes	280	4	69	89							
	MA	No	51	4	51	102		MA	No	210	4	210	98							
	QLC/B4	Yes	42	4	28	105		QLC^s	Yes	203	4	91	99							
	QLC/F4	No	52	4	21	96		CC/F4	Yes	293	4.75	2386	89							
	CC/F4	Yes	78	4.75	584	82														

designs according to one of their factors. Therefore, the 128-run k-factor QLC designs, say, cannot be constructed by concatenating two strength-3 designs with $k-1$ factors and adding the indicator factor. If it were possible to construct k-factor 128-run QLC designs by concatenating two 64-run strength-3 parents, then these parents should also involve k factors. Now, 64-run strength-3 parent designs involving 22 factors or more are necessarily even designs. Since concatenating two even designs results in a design that is even too, the QLC designs for 22 factors or more, which are even-odd, cannot be constructed by concatenating two strength-3 designs. For the 16-factor and 17-factor cases, even-odd parent designs do exist. If a construction by concatenation of the strength-3 QLC designs were possible, we should be able to extend the 16-factor and 17-factor designs with an extra factor that indicates the parent designs, and the extended designs should also have a strength of 3. We tried to extend the 16-factor and 17-factor designs using the algorithm of Schoen et al. (2010), but it did not produce an extension in 6 hours of computing time, while it normally takes less than a second to extend similar designs with 64 runs. For this reason, we conjecture that the 128-run QLC designs with 16 and 17 factors cannot be constructed by concatenating strength-3 designs. So, the reason why many of our CC/B4 designs are inadmissible is that we restrict ourselves to strength-3 parent designs and not that the CC/VNS algorithm performs poorly.

Table 6 shows that we were able to find designs with 18, 20, 21 and 30–33 factors which provide more degrees of freedom for 2FIs than the benchmark designs. The two 33-factor designs we obtained and one of our 21-factor designs are SOS designs. The same goes for the 28-factor design of Xu and Wong (2007) and the regular resolution IV designs with 25 and 29 factors.

The CC/VNS algorithm enabled us to add 11 admissible CC/B4 designs and 18 admissible CC/F4 designs to the literature on 128-run designs. We also found that certain designs from the literature are inadmissible when considering the generalized resolution, the B_4 value, the F_4 vector and the degrees of freedom for 2FIs.

5 Practical examples

We now return to the enzyme stability experiment and the software process simulation experiment that motivated this paper. For each of the two experiments, we explore several 64- and 128-run design options.

5.1 The enzyme stability experiment

The goal of the enzyme stability experiment was to improve the stability of an enzyme in a watery solution at room temperature. There were 17 experimental factors, which indicated the presence or absence of 17 possible additives. The experiment involved small test tubes with the enzyme and the additives. The tubes were stored at room temperature for eight weeks, and sampled at the start of the study and after 15, 30 and 60 days to check enzyme activity. A total of 64 combinations of the stabilizer was practically feasible, so that a 64-run design was used.

Table 7 shows 11 design options for the enzyme stability experiment. Designs with 64 runs available in the literature were the regular 64-run MA designs (Chen et al., 1993), the QLC designs (Xu and Wong, 2007), and the 17-factor design of Cheng et al. (2008), denoted as CMY in the table. We were reluctant to use the designs from the literature because they have 28 or more J_4-characteristics of 64, so that many pairs of 2FIs are completely aliased. Via an ad hoc procedure, we derived design X, which has only three J_4-characteristics of 64. Its complete F_4 vector is $F_4(64, 48, 32, 16) = (3, 6, 125, 625)$. So all the 64-run designs from the literature as well as design X have a generalized resolution of 4. Design X was the one eventually used for the experiment.

Table 7 includes three 64-run candidate designs for the enzyme stability experiment from our work for the present paper: the CC/B4 design, the CC/F4 design, and a design obtained by folding over the 32-run Paley matrix (design P in the table). It turns out that the 64-run CC/F4 design dominates design X, so that the ad hoc design actually used is inadmissible. The CMY design is also inadmissible because it is dominated by our CC/B4 design. In hindsight, we should perhaps have opted for the design obtained by folding over the 32-run Paley matrix, because of its large generalized resolution and the implication that the maximum J_4-characteristic value is only 16 for that design. The Paley-based design

Table 7: Design options for the 17-factor enzyme stability experiment.

N	Design	B_4	GR	$F_4^{\max}$	df
64	X	76	4	3	34
	CMY	60	4	28	46
	MA	59	4	59	43
	QLC/B4	59	4	59	43
	QLC/F4	64	4	40	43
	CC/B4	60	4	12	46
	CC/F4	65	4.5	83	46
	P	80	4.75	1286	31
128	QLC/B4	15	4	3	102
	QLC/F4	16	4.5	48	99
	CC/F4	28	4.75	189	78

provides 31 degrees of freedom for estimating 2FIs. This number compares rather poorly with the designs from the literature and with the CC/VNS designs. However, for the enzyme stability case, substantially fewer important interactions than 31 were expected, so that 31 degrees of freedom for estimating 2FIs would have been amply sufficient.

When discussing supersaturated designs, Marley and Woods (2010) argue that the degrees of freedom available for model selection should be at least half the number of eligible terms. As the strength-3 designs studied here are supersaturated for the 2FIs, heeding the advice of these authors in an unrestricted search for 2FIs (i.e., without imposing heredity) would require at least $\binom{17}{2}/2 = 68$ degrees of freedom for the 2FIs. Obviously, 64-run experiments are not large enough to provide this number of degrees of freedom. For this reason, we also study several admissible 128-run options that are compatible with the advice of Marley and Woods (2010). These design options are shown in the last three lines of Table 7. As minimizing the correlations among the 2FI contrast vectors reduces the ambiguity in the interpretation of the results, we prefer the 128-run CC/F4 design, despite the fact that it has the smallest number of degrees of freedom for estimating 2FIs of the three 128-run design options. By construction, the 128-run CC/F4 design has an indicator factor whose interactions with the 16 other factors can be estimated independently. As the total number of degrees of freedom for estimating interactions is 78 for that design, this leaves 62 degrees of freedom for the remaining 120 2FIs. This is compatible with the advice of Marley and Woods (2010).

5.2 The software process simulation experiment

The second motivating experiment is discussed in Houston et al. (2001) and is concerned with a sensitivity analysis of a simulation model for a software process. One of the designs used had 30 factors and 64 runs. Houston et al. (2001) mention that it was a regular design constructed using statistical software, but they do not provide any further details. Given that complete catalogs of regular 64-run designs of strength 3 have been available since 1993 (Chen et al., 1993), they might have used the 64-run 30-factor minimum aberration design. Table 8 shows the properties of that option, along with those of an alternative based on the folded-over 32-run Paley matrix and several admissible 128-run options.

Both 64-run design options listed for the software process simulation experiment are constructed by folding-over a strength-2 design. Both designs provide 31 degrees of freedom for estimating 2FIs and have a B_4 value of 945. However, the MA design is inadmissible because its GR value is smaller than that of the Paley-based design. So, the latter design is a better alternative for the software process simulation experiment. Given that there are 435 2FIs when 30 factors are studied, it is impossible to analyze the interactions without imposing restrictions such as strong heredity, which implies that a 2FI should be considered for inclusion in the fitted model only if both of its parent MEs are active. One option is to conduct the analysis of the MEs and the 2FIs in two successive steps, and to impose strong heredity restrictions in the second step (Miller and Sitter, 2001). The analysis would be compatible with the advice of Marley and Woods (2010) if the number of active MEs turns out to be at most 11.

Table 8 also lists three admissible design options with 128 runs. The number of degrees of freedom for estimating 2FIs of these designs is more than twice as large as that for

ACCEPTED MANUSCRIPT

Table 8: Design options for the 30-factor software process simulation experiment.

N	Design	B_4	GR	$F_4^{\max}$	df
64	MA	945	4	945	31
	P	945	4.75	15120	31
128	CC/B4	386	4	70	91
	MA	335	4	335	87
	CC/F4	396	4.75	3280	91

the listed 64-run options. Therefore, under strong heredity, model selection based on the 128-run designs would be compatible with the advice of Marley and Woods (2010) if the number of active MEs identified in the first step turned to be at most 19 (8 more than for the 64-run design options). Among the 128-run designs in Table 8, we would prefer CC/F4 design because it minimizes the correlations between pairs of 2FIs contrast vectors and because it has the largest number of degrees of freedom for estimating 2FIs.

6 Discussion

In this paper, we introduced the CC/VNS algorithm to optimize the concatenation of two strength-3 orthogonal arrays. The algorithm employs sign switches and column permutations to minimize the aliasing among the two-factor interactions in the concatenated design. Using the CC/VNS algorithm, we generated two-level even-odd designs with 64 and 128 runs and up to 33 factors. Sixteen out of the 18 newly generated 64-run designs and 29 out of the 36 newly generated 128-run designs were admissible in terms of the aliasing of two-factor interactions and in terms of the degrees of freedom for estimating two-factor interactions when compared with benchmark designs from the literature.

All but one of our 64-run designs have a smaller G-aberration than the designs of Chen et al. (1993), Block and Mee (2005), Xu and Wong (2007), and Xu (2009). We obtained the best 64-run designs in terms of the generalized resolution from projections

ACCEPTED MANUSCRIPT

from the folded-over 32-run Hadamard matrix given by the Paley construction. However, a drawback of these designs is that they are even. Therefore, they provide at most 31 degrees of freedom for estimating two-factor interactions. The even-odd 64-run designs we obtained by sequentially minimizing the F_4 vector for 9–11 factors have a better G-aberration than those based on the folded-over Paley matrix.

The 128-run designs we obtained by sequentially minimizing the F_4 vector for 16–33 factors have a better generalized resolution than all alternative designs available from the literature. We recommend the use of these designs when the experimenter's interest is in minimizing the correlation between pairs of two-factor interaction contrast vectors.

The indicator factor with $N/2$ positive ones and $N/2$ negative ones causes even concatenated designs, produced by concatenating two even parent designs, to become even-odd, and, for $k = N/4+1$ factors, to be second order saturated. When based on even-odd parent designs, concatenated designs can be even-odd without the inclusion of the indicator factor. Even-odd and SOS designs are attractive for estimating models including all the main effects and a considerable number of two-factor interactions. Alternative nonorthogonal designs to estimate interactions can be found in Li and Nachtsheim (2000), Smucker et al. (2011), Smucker et al. (2012), and Smucker and Drew (2015).

Selection strategies for models with main effects and interactions can be found in Draguljić et al. (2014), for instance. Alternatively, one might conduct a two-stage analysis similar to the one proposed by Miller and Sitter (2001). In the first stage of their proposed analysis, the active main effects are identified, while, in the second stage, only two-factor interactions obeying effect heredity are studied. An interesting subject for further research is to improve this approach by taking into account the independence of the two-factor interactions involving the indicator factor in our concatenated designs.

The indicator factor can also be used as a blocking factor to arrange the concatenated design in two blocks of size $N/2$. This blocking factor is orthogonal to the main effects and to all second-order interactions of the remaining factors. Thus, the upper parent design D_u and the optimal plan for the lower parent design D_l can be run on two different days or machines, offering more flexibility for the experimentation.

If the concatenated design is made up from non-isomorphic parent designs, we recommend to run the parent design with the smallest B_4 value first, or the one with the best F_4 vector, or the the largest number of degrees of freedom for estimating interactions, depending on the interest of the experimenter. There are four cases in which our 64-run designs are constructed from non-isomorphic parent designs and five cases in which our 128-run designs are constructed from non-isomorphic parents. Details are given in Supplementary Section D.

Finally, the CC/VNS algorithm may be able to improve on the designs with 64 and 128 runs of Xu and Wong (2007) in terms of G_2-aberration, by concatenating strength-2 parent designs instead of strength-3 parent designs. The main challenge here is to identify the ideal strength-2 parent designs. This would be an interesting topic for future research. Since our algorithmic approach is very general, it would also be interesting to investigate the concatenation of orthogonal arrays of different strengths and even different sizes. In addition, the parent designs considered could include nonorthogonal arrays, multi-level arrays or mixed-level arrays.

SUPPLEMENTARY MATERIALS

Supplementary sections.pdf Objective functions and fast update methods; algorithm performance evaluation; parent designs; concatenated designs; 128-run designs of strength 4.

Programs.zip Matlab implementation of the CC/VNS algorithm.

ACKNOWLEDGEMENTS

The research that led to this paper was financially supported by the Flemish Fund for Scientific Research FWO.

References

Avanthay, C., Hertz, A., and Zufferey, N. (2003). A variable neighborhood search for graph coloring. *European Journal of Operational Research*, 151:379–388.

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

Butler, N. A. (2004). Minimum G_2-aberration properties of two-level foldover designs. *Statistics & Probability Letters*, 67:121–132.

Butler, N. A. (2007). Results for two-level fractional factorial designs of resolution IV or more. *Journal of Statististical Planning and Inference*, 137:317–323.

Caporossi, G. and Hansen, P. (2004). Variable neighborhood search for extremal graphs. 5. Three ways to automate finding conjectures. *Discrete Mathematics*, 276:81–94.

Chen, J., Sun, D. X., and Wu, C. F. J. (1993). A catalogue of two-level and three-level fractional factorial designs with small runs. *International Statistical Review*, 61:131–145.

Cheng, C.-S., Mee, R. W., and Yee, O. (2008). Second order saturated orthogonal arrays of strength three. *Statistica Sinica*, 18:105–119.

Deng, L.-Y. and Tang, B. (1999). Generalized resolution and minimum aberration criteria for Plackett-Burman and other nonregular factorial designs. *Statistica Sinica*, 9:1071–1082.

Draguljić, D., Woods, D. C., Dean, A. M., Lewis, S. M., and Vine, A.-J. E. (2014). Screening strategies in the presence of interactions. *Technometrics*, 56:1–28.

Eglese, R. (1990). Simulated annealing: A tool for operational research. *European Journal of Operational Research*, 46:271–281.

Fleszar, K. and Hindi, K. S. (2004). Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, 155:402–413.

Garroi, J.-J., Goos, P., and Sörensen, K. (2009). A variable-neighbourhood search algorithm for finding optimal run orders in the presence of serial correlation. *Journal of Statistical Planning and Inference*, 139:30–44.

Glover, F. and Laguna, M. (1997). *Tabu Search*. Springer.

ACCEPTED MANUSCRIPT

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467.

Hansen, P., Mladenović, N., and Moreno Pérez, J. A. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6:319–360.

Hedayat, A., Sloane, N., and Stufken, J. (1999). *Orthogonal Arrays: Theory and Applications*. Springer.

Houston, D. X., Ferreira, S., Collofello, J. S., Montgomery, D. C., Mackulak, G. T., and Shunk, D. L. (2001). Behavioral characterization: finding and using the influential factors in software process simulation models. *The Journal of Systems and Software*, 59:259–270.

Kytöjoki, J., Nuortio, T., Bräysy, O., and Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers and Operations Research*, 34:2743–2757.

Li, W. and Lin, D. K. J. (2003). Optimal foldover plans for two-level fractional factorial designs. *Technometrics*, 45:142–149.

Li, W. and Lin, D. K. J. (2016). A note on foldover of 2^{k-p} designs with column permutations. *Technometrics*, 58:508–512.

Li, W., Lin, D. K. J., and Ye, K. Q. (2003). Optimal foldover plans for two-level nonregular orthogonal designs. *Technometrics*, 45:347–351.

Li, W. and Nachtsheim, C. (2000). Model-robust factorial designs. *Technometrics*, 42:345–352.

Marley, C. J. and Woods, D. C. (2010). A comparison of design and model selection methods for supersaturated experiments. *Computational Statistics and Data Analysis*, 54:3158–3167.

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

Mee, R. W., Schoen, E. D., and Edwards, D. J. (2017). Selecting an orthogonal or nonorthogonal two-level design for screening. *Technometrics*, 59:305–318.

Michalewicz, Z. and Fogel, D. (2004). *How to Solve It: Modern Heuristics*. Springer.

Miller, A. and Sitter, R. R. (2001). Using the folded-over 12-run Plackett-Burman design to consider interactions. *Technometrics*, 43:44–55.

Mladenović, N., Petrović, J., Kovačević-Vujčić, V., and Čangalović, M. (2003). Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search. *European Journal of Operational Research*, 151:389–399.

Sartono, B., Goos, P., and Schoen, E. (2015). Constructing general orthogonal fractional factorial split-plot designs. *Technometrics*, 57:488–502.

Schoen, E. D., Eendebak, P. T., and Nguyen, M. V. M. (2010). Complete enumeration of pure-level and mixed-level orthogonal arrays. *Journal of Combinatorial Designs*, 18:123–140.

Schoen, E. D. and Mee, R. W. (2012). Two-level designs of strength 3 and up to 48 runs. *Journal of the Royal Statistical Society Series C*, 61:163–174.

Sloane, N. J. A. (1999). A library of Hadamard matrices.

Smucker, B., del Castillo, E., and Rosenberg, J. (2011). Exchange algorithms for constructing model-robust experimental designs. *Journal of Quality Technology*, 43:28–42.

Smucker, B., del Castillo, E., and Rosenberg, J. (2012). Model-robust two-level designs using coordinate exchange algorithms and a maximin criterion. *Technometrics*, 54:367–375.

Smucker, B. and Drew, N. (2015). Approximate model spaces for model-robust experiment design. *Technometrics*, 57:54–63.

Sun, D. X., Li, W., and Ye, K. Q. (2008). Algorithmic construction of catalogs of non-isomorphic two-level orthogonal designs for economic run sizes. *Statistics and Application*, 6:144–158.

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

ACCEPTED MANUSCRIPT

Syafitri, U. D., Sartono, B., and Goos, P. (2015). I-optimal design of mixture experiments in the presence of ingredient availability constraints. *Journal of Quality Technology*, 47:220–234.

Tang, B. and Deng, L.-Y. (1999). Minimum G_2-aberration for nonregular fractional factorial designs. *The Annals of Statistics*, 27:1914–1926.

Wu, C. F. J. and Hamada, M. S. (2009). *Experiments: Planning, Analysis, and Optimization*. Wiley.

Xu, H. (2005). Some nonregular designs from the Nordstrom-Robinson code and their statistical properties. *Biometrika*, 92:385–397.

Xu, H. (2009). Algorithmic construction of efficient fractional factorial designs with large run sizes. *Technometrics*, 51:262–277.

Xu, H., Phoa, F. K., and Wong, W. K. (2009). Recent developments in nonregular fractional factorial designs. *Statistics Surveys*, 3:18–46.

Xu, H. and Wong, A. (2007). Two-level nonregular designs from quaternary linear codes. *Statistica Sinica*, 17:1191–1213.