POLITECNICO DI TORINO
Repository ISTITUZIONALE

Robust Designs Against Data Loss: A General Approach

Original
Robust Designs Against Data Loss: A General Approach / Fontana, R., Rapallo, F.. - 467:(2024), pp. 297-313. (SIS 2022 Caserta (Italy) June 22–24, 2022) [10.1007/978-3-031-65699-6_10].

Availability:
This version is available at: 11583/3016028 since: 2026-10-14T13:08:50Z

Publisher:
Springer

Published
DOI:10.1007/978-3-031-65699-6_10

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-65699-6_10

(Article begins on next page)

29 November 2026

Robust designs against data loss: A general approach

Roberto Fontana[0000−0002−3989−4887] and
Fabio Rapallo[0000−0001−6451−5140]

Abstract We describe an algorithm that, given an initial design $\mathcal{F}_n$ of size n and a linear model with p parameters, provides a sequence $\mathcal{F}_n \supset \ldots \supset \mathcal{F}_{n-k} \supset \ldots \supset \mathcal{F}_p$ of nested robust designs. The sequence is obtained by removing each individual run of $\mathcal{F}_n$ until a p-run saturated design $\mathcal{F}_p$ is obtained. The potential impact of the algorithm on real applications is high, because it can be used in a wide spectrum of designs. The initial fraction $\mathcal{F}_n$ can be of any type and the output sequence can be used to organize the experimental activity. The experiments can start with the runs corresponding to $\mathcal{F}_p$ and then continue by adding one run after the other (from $\mathcal{F}_{n-k}$ to $\mathcal{F}_{n-k+1}$) until the initial design $\mathcal{F}_n$ is obtained. In this way, if for some unexpected reasons the experimental activity has to be interrupted before the end when only $n-k$ runs have been completed, the corresponding $\mathcal{F}_{n-k}$ will have a high value of robustness for $k \in \{1, \ldots, n-p\}$. The algorithm uses the circuit basis, a special representation of the kernel of a matrix with integer entries.

Key words: Fractional factorial designs, D-optimality, incomplete designs.

1 Introduction

Optimal designs and orthogonal fractional factorial designs are frequently used in many fields of applications, including medicine, engineering and agriculture. They offer a valuable tool for dealing with problems where there are many factors involved and each run is expensive. The literature on the subject is extremely rich. A non-

Roberto Fontana
Department of Mathematical Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy e-mail: roberto.fontana@polito.it

Fabio Rapallo
Department of Economics, University of Genova, Via Vivaldi 5, 16126 Genova, Italy e-mail: fabio.rapallo@unige.it

exhaustive list of references includes [3] for design of experiments in general, [2], [7], and [18] for optimal designs and [15], [6], [14] for orthogonal fractional factorial designs.

When searching for an optimal experimental design, the aim is to select a design which produces the best estimates of the relevant parameters for a given sample size. There are many criteria for choosing an optimal design for the problem under study. They include alphabetical design criteria, and among these D-optimality is one of the most commonly used in applications.

In this work we focus on the notion of robustness of a design, [11]. Let us suppose that a given design has n runs and that the model to be estimated has p parameters, with $n > p$. The model-design pair determines the design matrix X. The notion of robustness is important mainly for two reasons. First, the robustness of the design can be interpreted as the probability that a randomly selected subset of p runs is a *saturated* design (i.e the p parameters of the model can be estimated). A high value of robustness has practical importance. If during the experimental activity $n-p$ runs are lost (i.e. the corresponding response values are not available) the probability that the p-run final design is *saturated* is high. Second, a close connection between robustness and D-optimality for a large class of model matrices is proved in [11].

The aim of this contribution is to provide a general order-of-the-runs criterion based on the notion of robustness. It can be applied to a wide range of possible designs. All designs for qualitative factors can be considered, from factorial designs with two-level factors, to multi-level and mixed-level factors. Even in some cases of continuous factors the proposed criterion can be applied. Since the robustness is based on the combinatorial properties of the design matrix, the unique assumption is to have a design matrix with integer entries. In practice combinatorial algorithms are limited by the dimension of the designs under analysis. We will discuss this issue in the next sections.

The main result of this work is an algorithm that starts from an initial design $\mathcal{F}_n$ (of size $n > p$) and removes each individual run of $\mathcal{F}_n$ until a p-run *saturated* design $\mathcal{F}_p$ is obtained. The choice of which point is removed at each step k ($1 \leq k \leq n-p$) aims to find a $(n-k)$-run sub-fraction $\mathcal{F}_{n-k}$ of the initial design with the highest value of robustness. The output of the algorithm is a sequence of *robust* fractions $\mathcal{F}_n \supset \ldots \supset \mathcal{F}_{n-k} \supset \ldots \supset \mathcal{F}_p$. In practice, the experimental activity can start with the runs corresponding to $\mathcal{F}_p$ and then continues by adding one run after the other (from $\mathcal{F}_{n-k}$ to $\mathcal{F}_{n-k+1}$) until the initial design $\mathcal{F}_n$ is obtained. In this way, if for some unexpected reasons the experimental activity has to be interrupted before the end when only $n-k$ runs have been completed, the corresponding $\mathcal{F}_{n-k}$ will have a high value of robustness, $1 \leq k \leq n-p$. It is worth noting that the value of robustness of $\mathcal{F}_{n-k}$ is high *for each* k, $1 \leq k \leq n-p$ as shown in the simulation study. In the simulation study, some examples are illustrated to prove the effectiveness of the algorithm. The problem of partial availability of data and techniques to prevent possible loss of information are addressed in, e.g., [4], [5], [19], both in model-based and model-free frameworks. In the paper [8] a combinatorial approach is introduced for the analysis of orthogonal arrays with removed runs using aberrations and the Generalized Word-Length Pattern criterion.

The algorithm introduced here uses the circuit basis, i.e., a special representation of the kernel of a matrix with integer entries. By exploiting some combinatorial properties of such a basis, we obtain a sequence of nested designs with high performance in terms of robustness. This result is obtained with a unique computation of the circuit basis in the first step. The theory of robustness based on circuits is fully described in [11], while the estimability of saturated designs using circuits is studied in [10].

It is worth noting that there are no restrictions on the way in which the initial design $\mathcal{F}_n$ is determined. It can be an orthogonal fractional factorial design, a D-optimal design, or any other design defined according to the user's preferences. The only restriction applies to the design matrix that must have integer entries. It follows that ANOVA-type models for qualitative variables can be considered. Polynomial models for continuous variables can also be considered with the restriction that the entries of the design are rational numbers. The examples below show that, from the combinatorial point of view, in some cases quantitative factors are easily rewritten in the qualitative framework. The general case of quantitative factors where an approximation of the design matrix is needed falls outside the scope of the present work, and we will provide some reflections for this in the concluding remarks.

The material is organized as follows. In Sect. 2 the definition of circuit basis is introduced together with its main algebraic and combinatorial properties, and the connections with robustness of a design are reviewed. Sect. 3 is devoted to the description of the proposed algorithm and some computational remarks. A first example on a small design is illustrated with full details. In Sect. 4 some examples are presented and discussed. Finally, Sect. 5 contains some final comments and pointers for future work.

2 Circuits and robustness

We consider a design $\mathcal{F}$ with n runs, chosen from a set $\mathcal{D}$ with N runs, $N > n$. For experiments with d discrete factors $X_1, \ldots, X_d$, the set $\mathcal{D}$ is usually represented as a Cartesian product such as

$$\{0, \ldots, s_1 - 1\} \times \cdots \times \{0, \ldots, s_d - 1\},$$

where $s_1, \ldots, s_d$ are the number of levels of the factors $X_1, \ldots, X_d$, respectively. However, for our theory the special coding of the factor levels and even the Cartesian product structure of the full-factorial design are irrelevant. We may simply assume that a subset of n runs has been selected from a large set labeled $\{1, \ldots, N\}$. In the language of fractional factorial designs, the design $\mathcal{F}$ is a fraction, while the large set $\mathcal{D}$ is a full-factorial design.

Given a full-factorial design $\mathcal{D}$ we consider a linear model on $\mathcal{D}$:

$$\mathbf{y} = X_{\mathcal{D}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{y}$ is the vector containing the response variable, $X_{\mathcal{D}}$ is the model matrix, $\boldsymbol{\beta}$ is the vector of parameters, and $\boldsymbol{\varepsilon}$ is the error term. Without loss of generality, to simplify some algebraic issues of our theory, we assume that the matrix $X_{\mathcal{D}}$ is full-rank with dimension $N \times p$, where p is the number of estimable parameters.

When a fraction $\mathcal{F}$ is selected, the expression of the model in Eq. (1) becomes

$$\mathbf{y} = X_{\mathcal{F}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\,, \tag{2}$$

where the model matrix $X_{\mathcal{F}}$ has dimension $n \times p$ and is obtained from $X_{\mathcal{D}}$ by selecting only the rows pertaining the chosen runs.

As pointed out in the Introduction, we aim at defining an algorithm which gives "optimal" subsets of a given fraction. Here, we use the criterion of robustness. Following [12] and [13], the robustness is defined in terms of saturated fractions.

Definition 1 Let $\mathcal{F}$ be a fraction with model matrix $X_{\mathcal{F}}$. The robustness of the fraction $\mathcal{F}$ under the model $X_{\mathcal{F}}$ is the proportion of saturated minimal fractions over the number of minimal fractions contained in $\mathcal{F}$:

$$r(X_{\mathcal{F}}) = \frac{\#\{\text{saturated } \mathcal{F}_p\}}{\binom{n}{p}}\,. \tag{3}$$

We observe that the number of runs of a minimal fraction is p and for a minimal fraction $\mathcal{F}$ the robustness can be either 0 or 1.

We now show how to use the circuits of the model matrix $X_{\mathcal{F}}$ to study the robustness of the fraction. Then, in the next section we will provide an algorithm to sequentially remove runs from $\mathcal{F}$ to maintain the robustness as high as possible.

To match the language of Combinatorics with the design theory, we work with the transposed of the model matrix, i.e., we consider the matrix $A_{\mathcal{F}} = X_{\mathcal{F}}^t$ and with a slight abuse of notation we still call it the model matrix. Note that working with $A_{\mathcal{F}}$ implies that the runs identify columns, while parameters identify rows.

In words, a circuit of $A_{\mathcal{F}}$ is an element $\mathbf{u}$ of $\ker(A_{\mathcal{F}})$ with integer entries and minimal support, where the support of a vector $\mathbf{u}$ is the set of indices i with $u_i \neq 0$. We denote by $\mathrm{supp}(\mathbf{u})$ the support of the vector $\mathbf{u}$.

Definition 2 Let $\mathbf{u} \in \mathbb{Z}^n$ be an n-dimensional integer vector. $\mathbf{u} \in \ker(A_{\mathcal{F}})$ is a circuit if the nonzero entries of $\mathbf{u}$ are relatively prime and there is no other vector $\mathbf{v} \in \ker(A_{\mathcal{F}})$ with $\mathrm{supp}(\mathbf{v}) \subset \mathrm{supp}(\mathbf{u})$.

The set of all the circuits of $A_{\mathcal{F}}$ is called the circuit basis of $A_{\mathcal{F}}$ and is denoted with $C(A_{\mathcal{F}})$. The circuit basis is always finite.

For a comprehensive introduction to circuits and its properties the reader can refer to [16] and [20]. Here, the key issue for using the circuit basis as a special basis of $\ker(A_{\mathcal{F}})$ is given by the following property.

Proposition 1 *Let $\mathcal{F}'$ be a sub-fraction of $\mathcal{F}$, and decompose each circuit of $A_{\mathcal{F}}$ into $\mathbf{u} = (\mathbf{u}_{\mathcal{F}'}, \mathbf{u}_{\mathcal{F}-\mathcal{F}'})$. The circuits of $A_{\mathcal{F}'}$ are*

$$\{\mathbf{u}_{\mathcal{F}'} \;:\; \mathbf{u} \in C(A_{\mathcal{F}}), \mathrm{supp}(\mathbf{u}) \subseteq \mathcal{F}'\}\,.$$

Prop. 1 says that the circuit basis is the natural representation of the kernel $\ker(A_{\mathcal{F}})$ when we need to remove runs, because we don't need to recompute the basis of the kernel at each step. The first computation contains all the information needed to compute the robustness also for all possible sub-fractions.

Moreover, the minimality property established in Def. 2 is used in the following result, from [10], which characterize the saturated minimal fractions using the circuit basis.

Proposition 2 *Let $A_{\mathcal{F}}$ be a model matrix with circuit basis $\mathcal{C}(A_{\mathcal{F}})$. A minimal sub-fraction $\mathcal{F}_p$, i.e., a fraction with p runs from $\mathcal{F}$, is saturated if and only if it does not contain any of the supports* $\operatorname{supp}(\mathbf{u})$ *for all* $\mathbf{u} \in \mathcal{C}(A_{\mathcal{F}})$.

Exploiting Prop. 2, in [11] an algorithm for finding robust fractions using the design points of a candidate set $\mathcal{D}$, based on an exchange-type strategy is described. Without introducing all the technical details, the algorithm in [11] is based on two general principles: (i) remove the point contained in the largest number of circuits; (ii) remove the point contained in the smallest circuits. Such two rules are summarized in the definition of a loss function $L(P)$ defined as

$$L(P) = \sum_{\operatorname{supp}(\mathbf{u}) \supseteq \mathcal{F}, \operatorname{supp}(\mathbf{u}) \ni P} \binom{n - \#\operatorname{supp}(\mathbf{u})}{p - \#\operatorname{supp}(\mathbf{u})}. \tag{4}$$

Note that the sum is taken over all the circuits $\mathbf{u}$ in the fraction containing the point P.

The algorithm is formed by the following main steps: (a) Start from a fraction $\mathcal{F}$; (b) Remove from $\mathcal{F}$ the run with the highest loss function; (c) Add a new run from $\mathcal{D}$ not in $\mathcal{F}$; (d) Repeat steps (b)-(c) until no reduction in the number of circuits is possible.

To actually compute the circuit basis $\mathcal{C}(A_{\mathcal{F}})$ for a given model matrix $A_{\mathcal{F}}$ there are several available packages and free software. For our computations we have used `4ti2`, see [1], a program for computing combinatorial objects like Markov bases, Graver bases, circuits, and more.

A common drawback of Algebraic Statistics tools is the limitation to small problems. The computation of the circuits does not make exception. The number of circuits for a model matrix on the full-factorial design increases fast with the number of runs and the computations are actually feasible only for small-sized problems. To give a rough idea, the circuits for a full-factorial 2^d design with main effects and first-order interactions is feasible only for $d \leq 6$. For the 2^d with only main effects, the circuit basis has 20 elements for $d = 3$; 1,348 elements for $d = 4$; 353,616 elements for $d = 5$.

As a running example, let us consider a $3 \cdot 2^2$ design with main effects and the interaction between the two binary factors. The full-factorial design has $N = 12$ points and there are $p = 6$ free parameters. To run the exchange-type algorithm described above, we compute the circuits of the model matrix on the full-factorial design and we obtain a circuit basis with 42 circuits: 18 circuits with support on 4 points and 24 circuits with support on 6 points. Now, to find a robust fraction with

a fixed size, say $n = 8$, it is enough to run the algorithm above with an arbitrary starting fraction with 8 runs. One can start with a randomly selected fraction, or can select a fraction satisfying some given criteria, such as D-optimality. However, the algorithm introduced in this work does not need the computation of the circuit basis for the full-factorial design, but only the circuit basis for the starting fraction, and therefore it can be applied also in relatively large examples.

We make explicit here some computational remarks, and we will come back on these issues later after the introduction of the new algorithm in Sect. 3. First, the procedure above is based on the loss function defined in Eq. (4). Such formula does not count the exact number of non-estimable minimal fractions and thus it is not assured that for each sample size it returns a fraction with the highest possible robustness. In the simulations described in [11], for sample sizes near to minimal the performance is quite good, but for large fractions, where the number of circuits contained in more than one minimal fraction is not negligible, the algorithm may yield a fraction with low robustness. However, in the next section we will illustrate why the chosen loss function is a good choice for the selection of the runs to be removed. Second, the algorithm assumes that the circuit basis for the candidate set (in general the full-factorial design) is available. The computation of the circuit basis for large designs can be actually unfeasible and thus the practical applicability is usually limited to small cases. Third, the algorithm above works for a fixed sample size, and it yields non-nested fractions when applied with different sample sizes. This is due to the random addition of a new run, and to the presence of ties, i.e., runs with the same loss function to be removed.

3 Removing runs from a fraction

In this section we consider another version of the circuit-based algorithm for removing runs from a given design. While the procedure described in the previous section was essentially based on the first property of the circuits, summarized in Prop. 2, the algorithm below fully exploits the second property of the circuits, described in Prop. 1.

Given a fraction $\mathcal{F}$ and a model with (transposed) design matrix $A_{\mathcal{F}}$, we use the circuit basis $C(A_{\mathcal{F}})$ as a "geometric tool" to choose the order of the points to be removed from $\mathcal{F}$ with the goal of obtaining the best possible robustness of the subfractions. Note that in the loss function in Eq. (4) only the circuits with support on p points or less are involved. Thus, we systematically remove from the circuit basis the circuits with support on more than p points. Moreover, to avoid computational problems we assume that the design matrix $A_{\mathcal{F}}$ is full-rank. The algorithm works as follows:

1. Start from a fraction $\mathcal{F}$ with n runs, and compute the circuit basis $C(A_{\mathcal{F}})$;
2. Compute the loss function for the runs in $\mathcal{F}$ based on $C(A_{\mathcal{F}})$;

3. Remove from $\mathcal{F}$ the run with the highest loss function $L(P)$. In case of ties, randomize among the runs with the highest loss function, and define a sub-fraction $\mathcal{F}'$ with $n-1$ runs;
4. Iterate items 2 and 3 until the desired number of runs has been removed.

The validity of the algorithm rests on two facts. First, from Prop. 1, the circuit basis $C(A_{\mathcal{F}})$ computed for the fraction $\mathcal{F}$ is valid also for all the sub-fractions $\mathcal{F}' \subset \mathcal{F}$, and thus the computation of the circuits is limited to the initial step and only for the given starting fraction. Second, the chosen expression of the loss function helps in reaching robust fractions at each step. Indeed, if we consider a circuit $\mathbf{u}$ with support contained in $\mathcal{F}$, the number of non-estimable fractions containing the circuit $\mathbf{u}$ and the design point P is

$$\widetilde{L}(P,u) = \binom{n - \#\mathrm{supp}(\mathbf{u})}{p - \#\mathrm{supp}(\mathbf{u})}$$

if $P \in \mathbf{u}$ and

$$\widetilde{L}(P,u) = \binom{n - \#\mathrm{supp}(\mathbf{u}) - 1}{p - \#\mathrm{supp}(\mathbf{u}) - 1}$$

if $P \notin \mathbf{u}$. The sum over all the circuits would yield a loss function of the form

$$\widetilde{L}(P) = \sum_{\mathrm{supp}(\mathbf{u}) \supseteq \mathcal{F}, \mathrm{supp}(\mathbf{u}) \ni P} \binom{n - \#\mathrm{supp}(\mathbf{u})}{p - \#\mathrm{supp}(\mathbf{u})} + \sum_{\mathrm{supp}(\mathbf{u}) \supseteq \mathcal{F}, \mathrm{supp}(\mathbf{u}) \not\ni P} \binom{n - \#\mathrm{supp}(\mathbf{u}) - 1}{p - \#\mathrm{supp}(\mathbf{u}) - 1}. \tag{5}$$

However, in view of our greedy strategy, we aim primarily at reducing the number of circuits because the surviving circuits entail new non-estimable minimal fractions in the subsequent steps. The idea of looking at the circuits instead of the non-estimable fractions rely on the idea that the circuits are the causes of the non-estimability and thus we concentrate in the elimination of such causes. As a consequence, the two addends in Eq. (5) have a totally different meaning because the second addend adds a positive contribution to the loss function for circuits not containing the data point P. Thus, we only keep the first sum and we use the loss function defined in Eq. (4).

Ideally the algorithm can be iterated until a saturated fraction is reached. Remember that in this case the robustness of the fraction can be either 0 or 1. Although robustness is useful for small designs with run size near to the minimum p, the algorithm works for all run sizes from n to p. Even in the intermediate cases, where the complete enumeration of all the sub-fractions may be computationally difficult, the proposed algorithm is able to easily find robust sub-fractions because it works on the runs and not on the sub-fractions.

Let us illustrate now a very small example in order to show the applicability of the algorithm above. We consider again the $3 \cdot 2^2$ case with main effects and the interaction between the two binary factors. The problem is to find robust sub-fractions of a D-optimal design. We use here a lexicographic order of the factor levels. In this example, we start with the fraction $\mathcal{F}_9$ which has $n = 9$ runs:

$$\mathcal{F}_9 = \{(-1,-1,+1), (-1,+1,-1), (-1,+1,+1), (0,-1,-1), \\ (0,+1,-1), (0,+1,+1), (+1,-1,-1), (+1,-1,+1), (+1,+1,-1)\} \tag{6}$$

with robustness $r(X_{\mathcal{F}_9}) = 0.5952$. The model has $p = 6$ parameters, so we seek for sub-fractions with 8, 7 and 6 runs.

There are 7 circuits with support contained in $\mathcal{F}_9$: 3 circuits with support on 4 points, 4 circuits with support on 6 points. We point out that the relevant circuit basis now has only 7 circuits, while the circuits basis for the full-factorial design has 42 elements, as described in the previous section. The 7 circuits are listed below (where the columns are ordered according to the list in Eq. (6)):

```
 0    0    0    1   -1    0   -1    0    1
 0    1   -1   -1    0    1    1    0   -1
 0    1   -1    0   -1    1    0    0    0
 1   -1    0   -1    1    0    1   -1    0
 1   -1    0    0    0    0    0   -1    1
 1    0   -1   -1    0    1    1   -1    0
 1    0   -1    0   -1    1    0   -1    1
```

In $\mathcal{F}_9$ the highest loss function $L(R)$ is reached by the 3 runs

$$(-1,+1,-1), (0,+1,-1), (+1,+1,-1)\,.$$

We randomly choose the first run from the above list, thus defining the robust sub-fraction

$$\mathcal{F}_8 = \{(-1,-1,+1), (-1,+1,+1), (0,-1,-1), (0,+1,-1), \\ (0,+1,+1), (+1,-1,-1), (+1,-1,+1), (+1,+1,-1)\} \tag{7}$$

with robustness $r(X_{\mathcal{F}_8}) = 0.7143$.

Among the 7 circuits above, only 3 of them still survive (the columns are ordered according to the list in Eq. (7)):

```
 0    0    1   -1    0   -1    0    1
 1   -1   -1    0    1    1   -1    0
 1   -1    0   -1    1    0   -1    1
```

To further reduce the sample size of the fraction we compute again the loss function of the 8 remaining points. The highest value is reached by the 4 runs:

$$(0,-1,-1), (0,+1,-1), (+1,-1,-1), (+1,+1,-1)\,.$$

We randomly choose the third run in this list, and we get

$$\mathcal{F}_7 = \{(-1,-1,+1), (-1,+1,+1), (0,-1,-1), (0,+1,-1), \\ (0,+1,+1), (+1,-1,+1), (+1,+1,-1)\} \tag{8}$$

with robustness $r(X_{\mathcal{F}_7}) = 0.8571$.

Only 1 circuit has support contained in $\mathcal{F}_7$ (the columns are ordered according the the list in Eq. (8)):

```
1    -1    0    -1    1    -1    1
```

In the last step to reach the minimal fraction, we remove one of the 6 runs in the support of the last circuit. We randomly choose the last run and we obtain

$$\mathcal{F}_6 = \{(-1,-1,+1),(-1,+1,+1),(0,-1,-1),\\ (0,+1,-1),(0,+1,+1),(+1,-1,+1)\}\,.$$

Of course this last fraction has robustness 1, because it is a minimal fraction and does not contain any support of the circuits.

In this example we can compare the robustness of the sub-fractions $\mathcal{F}_8$, $\mathcal{F}_7$, $\mathcal{F}_6$ with the robustness of all the possible sub-fractions with $8, 7, 6$ runs respectively. There are 9 sub-fractions of $\mathcal{F}_9$ with 8 runs: 6 of them have robustness 0.5357; 3 of them have robustness 0.7143. There are 36 sub-fractions of $\mathcal{F}_9$ with 7 runs: 3 of them have robustness 0 and they are actually non-estimable; 24 of them have robustness 0.5714; 9 of them have robustness 0.8571. Finally, there are 84 sub-fractions of $\mathcal{F}_9$ with 6 runs: 34 of them have robustness 0; 50 of them have robustness 1. We observe that our procedure has identified the highest robustness in all the steps.

Before the illustration of the examples in the next section, some computational remarks are needed to clarify the special features of the circuits for finding robust designs using our algorithm. First, the sub-fraction property of the circuits in Prop. 1 operates here in two ways. On one hand, we don't need the computation of the circuits for the full-factorial design, but only for the starting design. This makes feasible the computations also in cases where the circuits for the full factorial design cannot be computed. Moreover, the circuit basis in the first step is still valid throughout the whole algorithm, and no further computations are needed. These features allow us to use the algorithm also in intermediate-sized examples: using a standard PC, the time needed for the computation of the circuits for all the examples discussed in this paper ranges from 0.01 seconds for the running example above to 0.55 seconds for the example in Section 4.1.2.

4 Examples and applications

In Sect. 4.1 we evaluate the performance of the algorithm, that is the ability to produce a sequence of *robust* nested sub-fractions. We run a simulation study which considers different designs with both qualitative and quantitative factors. In Sect. 4.2, starting from the data available in a real application, we show the importance of choosing a robust fraction in terms of the ability to obtain a reliable estimate of β, the unknown vector of parameters, in the case some runs are lost.

4.1 Performance of the algorithm

We consider two examples. The examples have been chosen to show the possibility to use the proposed approach in different contexts. In each example we consider a starting design with $n > p$ runs. Then we analyze the robustness of its sub-fractions which are obtained removing $1, \ldots, n-p$ points by the starting design. It follows that by removing k points ($1 \leq k \leq n-p$), we must consider $\binom{n}{k}$ sub-fractions of the starting design and, for computing the robustness of each sub-fraction, we must consider $\binom{k}{p}$ size-p fractions which are contained in it. In this way we obtain the exact distribution of the robustness of all the sub-fractions of size $n-k$ of the starting design. This distribution allows us to evaluate the goodness of the solutions proposed by the algorithm. In the second example the number $\binom{n}{k}\binom{k}{p}$ becomes too large, and thus we build an approximation of the distribution of the robustness by sampling. More specifically we consider $\min(\binom{n}{k}, 1000)$ sub-fractions and we evaluate the robustness of each sub-fraction by classifying $\min(\binom{k}{p}, 1000)$ fractions of size p which are contained in it as saturated or not.

4.1.1 Example 1: a Plackett-Burman design

The first example considers five 2-level factors and a model with a constant term plus the 5 main effects. The number of degrees of freedom of the model is $p = 1 + 5 = 6$. The robustness of a Plackett-Burman design with $n = 12$ runs is analyzed. For this problem the circuit basis has 91 circuits, while the corresponding circuit basis for the full-factorial problem would consist of $44,560$ circuits. The values of the robustness of the fractions which are obtained removing $k = 1, \ldots, n-p = 6$ points are computed and compared with the values of the robustness corresponding to the fractions found by the algorithm. The distributions of the robustness for each $k = 1, \ldots, 6$ are exact. The case $k = 0$ (i.e. no points removed) provides the robustness of the initial design. The results are summarized in Fig. 1.

Table 1 compares the values of the robustness of the fractions found by the algorithm (r_*) with the 75th, 90th and 95th percentile of the distributions of the robustness (p_{75}, p_{90}, p_{95} respectively). It is worth noting that for each number k of points removed the algorithm provides values of robustness greater than the 75th percentile and apart from $k = 3$ equal to the 95th percentile.

4.1.2 Example 2: a design with quantitative continuous factors

The second example considers five continuous variables $x_i, i = 1, \ldots, 5$ which take values in the interval $[-1, +1]$. The model contains the intercept, five linear terms $x_i, i = 1, \ldots, 5$, five quadratic terms $x_i^2, i = 1, \ldots, 5$ and the four interaction terms $x_1x_2, x_1x_3, x_1x_4, x_1x_5$. The number of degrees of freedom of the model is $p = 1 + 5 + 5 + 4 = 15$. For this problem the circuit basis has 276 circuits. The

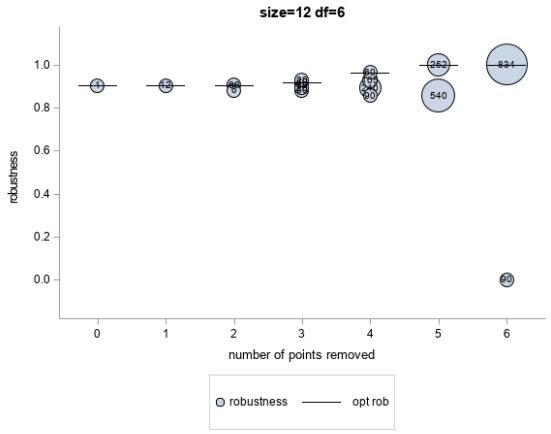

Fig. 1 Distribution of robustness *vs* number of points removed for the Plackett-Burman design in Sect. 4.1.1. For each sample size the robustness of the design selected by the proposed algorithm is represented with an horizontal line.

Table 1 Example 1: Comparison of the output of the algorithm (r_*) with the 75th, 90th, and 95th percentile of the distribution of the robustness (p_{75}, p_{90}, p_{95} respectively). The value k is the number of points removed by the initial design. The value corresponding to the robustness of the initial design (r_0) is given at $k = 0$.

k	p_{75}	p_{90}	p_{95}	r_*
0		r_0=0.903		
1	0.903	0.903	0.903	0.903
2	0.905	0.905	0.905	0.905
3	0.917	0.917	0.929	0.917
4	0.929	0.964	0.964	0.964
5	1	1	1	1
6	1	1	1	1

robustness of a D-optimal design with with $n = 20$ runs is analyzed. The 20-run D-optimal design has been obtained using as candidate set the full factorial design $\{-1, +1\}^5$ The values of the robustness of the fractions which are obtained removing $k = 1, \ldots, n - p = 5$ points are computed and compared with the values of the robustness corresponding to the fractions found by the algorithm. The distributions of the robustness for each $k = 1, \ldots, 5$ are obtained by sampling. The results are summarized in Fig. 2.

Table 2 compares the values of the robustness of the fractions found by the algorithm (r_*) with the 75th, 90th and 95th percentile of the distributions of the robustness (p_{75}, p_{90}, p_{95} respectively). It is worth noting that for each number k of points removed by the initial design the algorithm provides values of robustness greater than the 95th percentile.

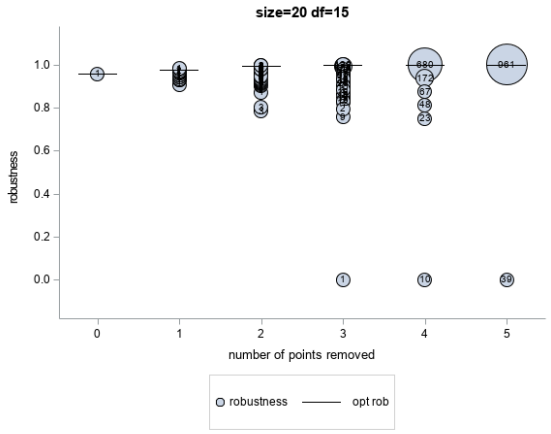

Fig. 2 Sampling distribution of robustness *vs* number of points removed for the 5-factor example in Sect. 4.1.2. For each sample size the robustness of the design selected by the proposed algorithm is represented with an horizontal line.

Table 2 Example 2: Comparison of the output of the algorithm (r_*) with the 75th, 90th, and 95th percentile of the distribution of the robustness (p_{75}, p_{90}, p_{95} respectively). The value k is the number of points removed by the initial design. The value corresponding to the robustness of the initial design (r_0) is given at $k = 0$.

k	p_{75}	p_{90}	p_{95}	r_*
0		r_0=0.954		
1	0.971	0.975	0.977	0.978
2	0.979	0.989	0.991	0.994
3	0.993	0.996	1	1
4	1	1	1	1
5	1	1	1	1

4.2 The impact of data loss in a real-data application

We consider the application described in [17]. Briefly, a new technique for approximating the stress in pad-type nozzles attached to a spherical shell is presented. The stress values corresponding to a single replicated full factorial design with three 3-level factors (A,B, and C) are used for studying, using standard ANOVA, the effects of the factors on the membrane stress ratio. For showing the impact that our algorithm could have in practical applications we use the discontinous membrane stress (approximate analysis) as response. The full factorial design $\mathcal{D} = \{-1, 0, 1\}^3$ with the corresponding values of the response ($\mathbf{y}$) are reported in Table 3.

As in [17], we consider a model with main effects and two-order interactions. The model can be written in matrix form as in Eq. (1), i.e.

Table 3 Values of the discontinous membrane stress y observed for each run of the full factorial design $\mathcal{D}$.

Case no.	A	B	C	$\mathbf{y}$	Case no.	A	B	C	$\mathbf{y}$
1	-1	-1	-1	191.8	15	1	0	0	278.3
2	0	-1	-1	230.5	16	-1	1	0	153.7
3	1	-1	-1	269.6	17	0	1	0	213.3
4	-1	0	-1	159.3	18	1	1	0	236
5	0	0	-1	175.5	19	-1	-1	1	208.6
6	1	0	-1	186.3	20	0	-1	1	293.6
7	-1	1	-1	154.8	21	1	-1	1	369.8
8	0	1	-1	167.6	22	-1	0	1	170.5
9	1	1	-1	154.8	23	0	0	1	234
10	-1	-1	0	227.5	24	1	0	1	286
11	0	-1	0	295.1	25	-1	1	1	152.4
12	1	-1	0	369	26	0	1	1	206.1
13	-1	0	0	174.6	27	1	1	1	251.9
14	0	0	0	231.5					

$$\mathbf{y} = X_{\mathcal{D}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The row $x_{(a,b,c)}$ of $X_{\mathcal{D}}$ corresponding to the design point $(a, b, c) \in \mathcal{D}$ is defined as

$$(1, x_a, x_b, x_c, x_{ab}, x_{ac}, x_{bc})$$

where $x_y = (1_{y=-1} - 1_{y=1}, 1_{y=0} - 1_{y=1})$, $x_{yz} = x_y \otimes x_z$, $y, z \in \{a, b, c\}$, $\otimes$ denotes the Kronecker product, and 1_c is the indicator function that gives 1 if the condition c is true and 0 otherwise. The number of columns of $X_{\mathcal{D}}$ i.e. the number of degrees of freedom of the model is $p = 1 + 3 \cdot 2 + 3 \cdot 4 = 19$. The least-square estimate $\hat{\beta}$ of β is obtained as

$$\hat{\beta} = (X_{\mathcal{D}}^t X_{\mathcal{D}})^{-1} X_{\mathcal{D}}^t \mathbf{y}. \tag{9}$$

Using our algorithm we obtain a sequence of robust fractions $\mathcal{F}_{27} \equiv \mathcal{D} \supset \ldots \supset \mathcal{F}_{19}$. We focus on $\mathcal{F}_{25}, \mathcal{F}_{24}$, and $\mathcal{F}_{23}$ which are the fractions of $\mathcal{D}$ which correspond to the loss of $2, 3$, and 4 runs, respectively. More specifically $\mathcal{F}_{25} = \overline{\mathcal{D}}_{\{18,1\}}$, $\mathcal{F}_{24} = \overline{\mathcal{D}}_{\{18,1,23\}}$, and $\mathcal{F}_{23} = \overline{\mathcal{D}}_{\{18,1,23,25\}}$, where $\overline{\mathcal{D}}_I$ means the full factorial design without the *case no.* in the set I, where the *case no.* of each run of the full factorial $\mathcal{D}$ is defined in Table 3.

Using $\mathcal{F}_x$ we compute the corresponding estimate $\hat{\beta}_x$ of β, $x = 23, 24, 25$. We measure how far $\hat{\beta}_x$ is from $\hat{\beta}$ (the estimate of β obtained using the full factorial design $\mathcal{D}$) using the mean error e_x defined as

$$e_x = \frac{1}{p} \sum_{i=1}^{p} \left| \frac{(\hat{\beta}_x)_i - (\hat{\beta})_i}{(\hat{\beta})_i} \right| = \frac{1}{19} \sum_{i=1}^{19} \left| \frac{(\hat{\beta}_x)_i - (\hat{\beta})_i}{(\hat{\beta})_i} \right|$$

where $(v)_i$ denotes the i-th component of the vector v.

Table 4 For each number of available runs, $x = 23, 24, 25$, the mean error e_x, the number of estimable fractions E_x among all the sub-fractions of size x of $\mathcal{D}$, the relative frequency of the event $e_x^k > e_x$, and the relative frequency of non-estimable fractions $\overline{E}_x$ are reported.

x	e_x	E_x	$\#(e_x^k > e_x)/E_x$	$\overline{E}_x/\binom{27}{x}$
23	0.299	16,821	0.838	0.0415
24	0.198	2,898	0.876	0.0092
25	0.269	351	0.567	0

For $x = 23, 24, 25$ we build all the fractions $\mathcal{A}_x^k$ of size x of $\mathcal{D}$, $k = 1, \ldots, \binom{27}{x}$. For each fraction $\mathcal{A}_x^k$ we compute $\hat{\beta}_x^k$, the least-square estimates of β using $\mathcal{A}_x^k$, and the corresponding errors e_x^k. It is worth noting (see Table 4) that

- the errors e_x are less than the median of e_x^k for all the x and for $x = 23, 24$ e_x are close to the 15-th percentiles of e_x^k, meaning that approximately 85% of fractions $\mathcal{A}_x^k$ provide mean errors e_x^k greater than e_x;
- for $x = 23$, and $x = 24$ there are $\overline{E}_{23} = 729$, and $\overline{E}_{24} = 27$ non-estimable fractions, and $E_{23} = 16,821$, and $E_{24} = 2,898$ estimable fractions, respectively (a fraction $\mathcal{F}$ is non-estimable when $\det(X_{\mathcal{F}}^t X_{\mathcal{F}}) = 0$). It follows that the probability that a randomly chosen fraction is non-estimable is $729/17,550 \approx 4.15\%$ for $x = 23$ and $27/2925 \approx 0,92\%$ for $x = 24$.

We run a simulation study for assessing the stability of the results. We considered $N = 10,000$ vectors $\mathbf{y_i}$ generated as

$$\mathbf{y_i} = X_{\mathcal{D}}\boldsymbol{\beta} + \boldsymbol{\varepsilon_i},\ i = 1, \ldots, N$$

where $\boldsymbol{\beta}$ is taken equal to $\hat{\beta}$ as defined in Eq. (9), $\boldsymbol{\varepsilon_i}$ is a vector of independent standard normally distributed random variables, and $\boldsymbol{\varepsilon_i}$ and $\boldsymbol{\varepsilon_j}$ are independent when $i \neq j$. For each simulation i we compute $e_{i,x}$, the error obtained using the robust fraction $\mathcal{F}_x$ and $\tilde{e}_{i,x}$ the error obtained using a randomly chosen fraction of $\mathcal{D}$ of size x, $x = 23, 24, 25$. The empirical distribution functions of $\{e_{i,25} : i = 1, \ldots, 10,000\}$ and $\{\tilde{e}_{i,25} : i = 1, \ldots, 10,000\}$ computed using $x = 25$ runs (that is two runs have been lost) are reported in Fig. 3. It is evident that the errors obtained using the robust fraction $\mathcal{F}_x$ are smaller than those obtained using a random fraction of size x of $\mathcal{D}$ (e.g. the median of $\{e_{i,25} : i = 1, \ldots, 10,000\}$ is 0.0647 and the median of $\{\tilde{e}_{i,25} : i = 1, \ldots, 10,000\}$ is 0.0854). The goodness of the results is also confirmed for $x = 23$ and 24.

5 Final remarks

The main result of this contribution is an algorithm for organizing the n runs of a given fraction $\mathcal{F}_n$ in such a way that, if for some reasons k of the n runs are lost, the remaining $n - k$ runs constitute a *robust* design, $1 \leq k \leq n - p$. As shown in the

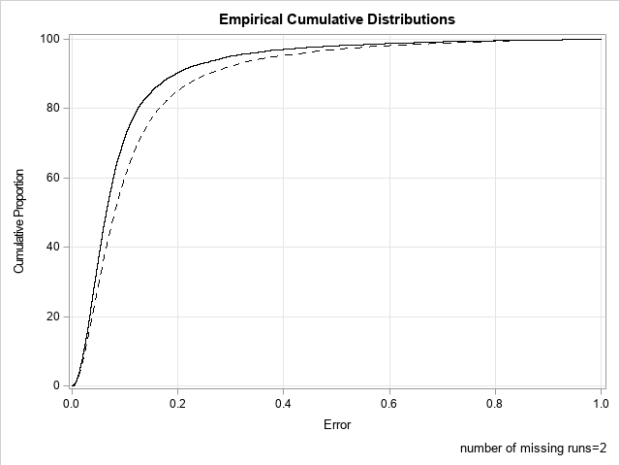

Fig. 3 Empirical distribution functions of $\{e_{i,25} : i = 1, \ldots, 10,000\}$ (solid line) and $\{\tilde{e}_{i,25} : i = 1, \ldots, 10,000\}$ (dashed line).

simulation study, the algorithm can provide very good designs in terms of robustness for all $k \in \{1, \ldots, n-p\}$. This means that k does not need to be defined at the design stage (it would have been extremely difficult to make a hypothesis on the number k of runs that could be lost before starting the execution of the experiments).

The algorithm can be used with any type of initial design. The starting design $\mathcal{F}_n$ can be an orthogonal fractional factorial design or a D-optimal design or any user-defined design. It is worth noting that the algorithm can work in many practical siyuations. The reason is that the circuits needed are those of the matrix $A_{\mathcal{F}_n}$ which has dimension $p \times n$ and, usually in the applications, both the number of parameters p and the size of the starting fraction $\mathcal{F}_n$ are not large.

One of the requirements is that the matrix $A_{\mathcal{F}_n}$ must have integer values. This is always the case for models with qualitative factors. Some preliminary work is needed for models with quantitative variables. For a model matrix with real numbers it is possible to build an approximate version of it with rational entries. The approximation can be built as accurately as required since the set of rational numbers $\mathbb{Q}$ is dense in the set of real number $\mathbb{R}$. Finally the model matrix can be transformed into a matrix with integer values simply by multiplying the rational matrix by a suitable integer constant. In some cases, as shown in the second example of Sect. 4, a D-optimal design used as a starting design for the algorithm contains only points with integer entries, and from the combinatorial point of view the problem reduces immediately to the qualitative case. Although the approximation of a real design matrix with a rational one is outside the scope of the present work, some first experiments in this direction are promising. For instance, we have considered the design matrix with non-rational entries presented in [9], page 1674. Using 1, 2, or 3 decimal places to approximate the real entries, we obtain in all cases the same structure of the circuit basis.

Acknowledgements The authors are members of INdAM–GNAMPA.

References

1. 4ti2 team: 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. `https://4ti2.github.io` (2018)
2. Atkinson, A., Donev, A., Tobias, R.: Optimum experimental designs, with SAS, vol. 34. Oxford University Press, Oxford, UK (2007)
3. Bailey, R.A.: Design of comparative experiments, vol. 25. Cambridge University Press, Cambridge, UK (2008)
4. Butler, N.A., Ramos, V.M.: Optimal additions to and deletions from two-level orthogonal arrays. J. R. Stat. Soc. Ser. B. Stat. Methodol. **69**(1), 51–61 (2007). DOI 10.1111/j.1467-9868.2007.00576.x
5. Dey, A.: Robustness of block designs against missing data. Stat. Sin. **3**(1), 219–231 (1993)
6. Dey, A., Mukerjee, R.: Fractional Factorial Plans. John Wiley & Sons, New York (2009)
7. Fedorov, V.V.: Theory of optimal experiments. Elsevier, United States (2013)
8. Fontana, R., Rapallo, F.: On the aberrations of mixed level orthogonal arrays with removed runs. Stat. Pap. **60**(2), 479–493 (2019). DOI 10.1007/s00362-018-01069-5
9. Fontana, R., Rapallo, F.: Combinatorial analysis of factorial designs with ordered factors. In: Balzanella, A., Bini, M., Cavicchia, C., Verde, R. (eds.) SIS 2022 Book of the Short Papers, pp. 1670–1675. Pearson (2022).
10. Fontana, R., Rapallo, F., Rogantin, M.P.: A characterization of saturated designs for factorial experiments. J. Stat. Plan. Inference **147**, 205–211 (2014). DOI 10.1016/j.jspi.2013.10.011
11. Fontana, R., Rapallo, F., Wynn, H.P.: Circuits for robust designs. Stat. Pap. **63**(5), 1537–1560 (2022). DOI 10.1007/s00362-021-01285-6
12. Ghosh, S.: On robustness of designs against incomplete data. Sankhya: The Indian Journal of Statistics, Series B (1960-2002) **40**(3/4), 204–208 (1979)
13. Ghosh, S.: Robustness of bibd against the unavailability of data. J. Stat. Plan. Inference **6**(1), 29–32 (1982). DOI 10.1016/0378-3758(82)90053-2
14. Hedayat, A.S., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays: Theory and Applications. Springer, New York (2012)
15. Mukerjee, R., Wu, C.F.J.: A Modern Theory of Factorial Design. Springer, New York (2007)
16. Ohsugi, H., Hibi, T.: Toric ideals and their circuits. J. Commut. Algebra **5**(2), 309–322 (2013). DOI 10.1216/JCA-2013-5-2-309
17. Oikawa, T., Oka, T.: A new technique for approximating the stress in pad-type nozzles attached to a spherical shell. J. Pressure Vessel Technol. **109**(2), 188–192 (1987). DOI 10.1115/1.3264894
18. Pukelsheim, F.: Optimal design of experiments. SIAM, New York (2006)
19. Street, D.J., Bird, E.M.: D-optimal orthogonal array minus t run designs. J. Stat. Theory Pract. **12**(3), 575–594 (2018). DOI 10.1080/15598608.2018.1441081
20. Sturmfels, B.: Gröbner bases and convex polytopes, *University Lecture Series*, vol. 8. American Mathematical Society, Providence, RI (1996)