Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces

Laurent Gilquin, Clémentine Prieur, Elise Arnaud

Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces

Laurent Gilquin, Clémentine Prieur, Elise Arnaud

INRIA

LJK, Université de Grenoble, BP53 38041 Grenoble cedex, France

Abstract

This paper deals with the estimation of grouped Sobol' indices in the case where the dependence inside each group is given by sets of linear ordered constraints. In the framework of independent inputs, the replication method allows to estimate first-order indices with only two designs. Through the use of orthogonal arrays of strength two the replication method can be applied to estimate closed second-order indices. We extend this methodology to estimate first-order and closed second-order grouped Sobol' indices under sets of linear constraints. The construction of the two designs required by the replication method is now based on the simplex geometric structure to handle the constraints within each set. We propose a space filling strategy to construct these designs.

Keywords: sensitivity analysis, grouped Sobol' index, dependence, replicated designs, simplex

1. Introduction

Sensitivity analysis studies how the uncertainty on an output of a mathematical model can be attributed to sources of uncertainty among the inputs. There are two main classes of sensitivity analyses called local and global sensitivity analysis. The former addresses sensitivity relatively to a nominal value of a given parameter. The latter examines sensitivity on the whole set of variation of the parameter. Global sensitivity analysis of complex and

Email address: laurent.gilquin@inria.fr (Laurent Gilquin)

expensive mathematical models is a common practice to identify influent inputs and detect the potential interactions between them. Among the large number of available approaches, the variance-based method introduced by Sobol' [1] allows to calculate sensitivity indices called Sobol' indices. Each index gives an estimation of the influence of an individual input or a group of inputs. These indices give an estimation of how the output uncertainty can be apportioned to the uncertainty in the inputs. One can distinguish first-order indices that estimate the main effect from each input or group of inputs from higher-order indices that estimate the corresponding order of interactions between inputs. Closed k-th order indices estimate k-th order interactions in addition to the main effect of each of the k inputs. This estimation procedure requires a significant number of model runs, number that has a polynomial growth rate with respect to the input space dimension. This cost can be prohibitive for time consuming models and only a few number of runs is not enough to retrieve accurate informations about the model inputs. Saltelli [2] proposed an improvement to reduce the number of runs but the total cost still depends linearly on the dimension of input space.

The notion of replicated designs to estimate first-order Sobol' indices probably goes back to McKay [3] and appears later in Mara *et al.* [4]. These last authors combine replicated designs with "pick-freeze" estimators [1] to estimate first-order Sobol' indices. The procedure in Mara *et al.* [4] has the major advantage of reducing drastically the estimation cost as the number of runs becomes independent of the input space dimension. This procedure has been further deeply studied and generalized in Tissot *et al.* [5] to the estimation of closed second-order indices. The generalization to closed second-order Sobol' indices relies on the replication of randomized orthogonal arrays (see Lemieux [6] or Owen [7]).

The motivation of our paper is to extend this methodology in presence of dependent inputs. Indeed, the case of correlated parameters has to be tackled with caution, as the calculation of single input indices does not provide anymore a proper information, that can be easily interpreted. One strategy is thus to define grouped indices for groups of correlated variables as proposed in Jacques *et al.* [8]. However, up to our knowledge, the problem of estimating those grouped indices, with a reasonable computing cost, has not been addressed in the literature. In this paper, we address this issue by proposing an approach based on replicated designs and randomized orthogonal arrays that enables to take into account dependency within inputs. We suppose that this dependency can be expressed through constraints. This approach can be

used facing any set of constraints at the condition that one is able to provide points in the input space that verify the considered constraints. Guided by our application on a land-use and transport integrated model (LUTI) where some economical parameters are linked by order relations, this paper focus on the case of sets of linear ordered constraints. Thus we propose a sampling strategy based on the simplex geometric structure, that ensures a proper input space filling.

This paper is organized as follows. Section 2 recalls the notion of grouped Sobol' indices (see Jacques *et al.* [8]) to study the sensitivity of an output to groups of inputs where variables within a group are dependent but variables of different groups are independent. In Section 3 we review the use of replication method and randomized orthogonal array to estimate first and closed second-order Sobol' indices. Section 4 is devoted to the sampling strategy adapted to our sets of linear ordered constraints and section 5 gives a summary of the whole procedure. Numerical experimentations are conducted in section 6. Classical case studies are addressed to compare our method to the ones in Sobol' [1] or Saltelli [2].

2. Definition of Sobol' indices

2.1. Sobol' indices for independent inputs

Consider the following model defined from a black box perspective:

$$f\colon \left\{ \begin{array}{ccc} \mathbb{R}^d & \to & \mathbb{R} \\ x=(x_1,\ldots,x_d) & \mapsto & y=f(x) \end{array} \right. \tag{1}$$

where y is the output of the model f, x the input vector.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We model the uncertainty on the inputs by a random vector $X=(X_1,\ldots,X_d)$ whose components are independent. Let $P_X = P_{X_1}\otimes\ldots\otimes P_{X_d}$ denote the distribution of X. We assume that $f \in \mathbb{L}^2(P_X)$. The model f can then be uniquely decomposed into summands of increasing dimensions (functional ANOVA decomposition [1, 9]):

$$f(X) = f_0 + \sum_i f_i(X_i) + \sum_{i<j} f_{ij}(X_i,X_j) + \cdots + f_{1\ldots d}(X_1,\ldots,X_d)\ , \tag{2}$$

where each component of the decomposition verifies:

$$\int f_{i_1\ldots i_s}(x_{i_1},\ldots,x_{i_s}) dP_{X_{i_k}}(x_{i_k}) = 0,\ \forall k \in \{1,\ldots,s\}, \forall i_1,\ldots,i_s \in \{1,\ldots,d\}.$$

This implies that $f_0 = \mathrm{E}[Y]$ and that the components are mutually orthogonal with respect to P_X. Let $I \subseteq \{1, \ldots, d\}$, each component is defined by:

$$f_I(X_I) = \mathrm{E}[Y|X_I] - \sum_{J \subset I} f_J(X_J).$$

The functional decomposition can be used to measure the global sensitivity of the output Y to the input X_i. By squaring and integrating (2), due to orthogonality we get:

$$V = \sum_i V_i + \sum_{i<j} V_{ij} + \cdots + V_{1,\ldots,d}$$

where $$V_I = \mathrm{Var}[f_I(X_I)] = \mathrm{Var}[\mathrm{E}[Y|X_I]] - \sum_{J \subset I} V_J$$

and $$V = \mathrm{Var}[Y]\,.$$

Resulting from this decomposition, the Sobol' indices are defined by:

$$S_I = \frac{V_I}{V}\,.$$

Let $|I|$ denote the cardinal of I. This index measures the contribution to V of $|I|^{\text{th}}$-order interaction between the X_i, $i \in I$.

Closed Sobol' indices are defined by:

$$\underline{S_I} = \frac{\mathrm{Var}[\mathrm{E}[Y|X_I]]}{V}\,.$$

The closed Sobol' index $\underline{S_I}$ measures the contribution of the X_i, $i \in I$ by themselves or in interaction with each other.

As an example, if there exist $i \neq j \in \{1, \ldots, d\}$ such that $I = \{i, j\}$, then $\underline{S}_{ij} = S_{ij} + S_i + S_j$.

At last, note that $\sum_{I \subset \{1,\ldots,d\}, I \neq \varnothing} S_I = 1$, allowing a direct interpretation of the value of each index.

2.2. Sobol' indices for dependent inputs

Consider again the model (1). We now suppose that X is a vector of d inputs among which one or multiple groups of variables are correlated. Variables belonging to the same group are dependent but variables of different groups are independent. This implies that one variable can not appear in two groups. Each group is denoted with a multidimensional variable. X can then be defined as follows:

$$\begin{aligned} X &= (X_1, \ldots, X_I, \underbrace{X_{I+1}, \ldots, X_{I+k_1}}_{\vec{X}_{I+1}}, \ldots, \underbrace{X_{I+1+\Sigma_{i-1}}, \ldots, X_{I+\Sigma_i}}_{\vec{X}_{I+i}}, \\ &\quad \ldots, \underbrace{X_{I+k_1+k_2+\cdots+k_{m-1}+1}, \ldots, X_d}_{\vec{X}_{I+m}}) \text{ where } \Sigma_i = \sum_{l=0}^{i} k_l,\ k_0 = 0, \quad (3) \end{aligned}$$

each $\vec{X}_{I+i},\ i \in \{1, \ldots, m\}$ contains k_i variables.

When inputs are dependent, equation (2) given by the FANOVA decomposition of f does not hold as the components of the decomposition are not mutually orthogonal. The correlation of variables within $\vec{X}_{I+1}$ implies that a part of the sensitivity of one of these variables is explained by the other correlated variables. We can still compute the closed Sobol' index associated to each variable (or subset) of $\vec{X}_{I+1}$ but its value is difficult to interpret. An alternative is to define grouped Sobol' indices as in Jacques *et al.* [8]. These indices are defined like in the classical case of independent inputs, but refer to the multidimensional variables. We consider X defined as in (3). First-order grouped Sobol' indices are defined by:

$$S_j = \frac{\mathrm{Var}[\mathrm{E}[Y|\vec{X}_j]]}{V}\ ,\ \forall j \in \{1, \ldots, I+m\}\ .$$

If $\vec{X}_j$ is one-dimensional, then $\vec{X}_j = X_j$ and S_j has the same expression as in the independent case. If $\vec{X}_j$ is multidimensional (case $j \in \{I+1, \ldots, I+m\}$), $\vec{X}_j = \vec{X}_{I+l},\ l \in \{1, \ldots, m\}$, the index is:

$$S_j = \underline{S}_{I+1+\Sigma_{l-1},\ldots,I+\Sigma_l} = \frac{\mathrm{Var}[\mathrm{E}[Y|X_{I+1+\Sigma_{l-1},\ldots,I+\Sigma_l}]]}{V}\ .$$

This index measures the impact of $\vec{X}_j$ on the output Y. Higher order Sobol' indices can also be defined. In particular, second-order grouped Sobol' indices

are given by:

$$S_{j,k} = \frac{\mathrm{Var}[\mathrm{E}[Y|\vec{X}_j, \vec{X}_k] - \mathrm{E}[Y|\vec{X}_j] - \mathrm{E}[Y|\vec{X}_k]]}{V} \ .$$

This index measures the impact of the interaction between $\vec{X}_j$ and $\vec{X}_k$ on the output Y. The grouped Sobol' indices allow us to estimate the sensitivity of our model to the sets of linear ordered constrained inputs and the remaining unidimensional independent inputs.

2.3. standard estimation of Sobol' indices

We define by design a $n \times d$ matrix which consists in n evaluations of each $X_i,\ i = 1, \ldots, d$. The standard estimation of a closed Sobol' index $\underline{S_I}$ proposed by Sobol' requires $2n$ evaluations of the model through two designs [1]. Denoting by I^c the complement of $I \subseteq \{1, \ldots, d\}$, the second design of n evaluations is created from the first one by re-sampling the columns indexed by I^c. This method is referred as the Sobol' method. Let Y and Y_I be the vectors of evaluations associated to those two designs. The expression of $\underline{S_I}$ can be rewritten as follows:

$$\underline{S_I} = \frac{\mathrm{E}[YY_I] - \mathrm{E}[Y]\mathrm{E}[Y_I]}{\mathrm{Var}[Y]} \ .$$

An efficient estimator of $\underline{S_I}$ with good asymptotic properties [10, 11] is:

$$\widehat{\underline{S_I}} = \frac{\frac{1}{n}\sum_{j=1}^{n} Y^j Y_I^j - \left(\frac{1}{n}\sum_{j=1}^{n} Y^j\right)\left(\frac{1}{n}\sum_{j=1}^{n} Y_I^j\right)}{\frac{1}{n}\sum_{j=1}^{n} (Y^j)^2 - \left(\frac{1}{n}\sum_{j=1}^{n} (Y^j)^2\right)} \ . \tag{4}$$

Note that this method can be applied to independent inputs or independent groups of inputs. Its main drawback is the increasing number of needed evaluations. Estimating all first-order, respectively all closed second-order, Sobol' indices requires $n(d+1)$, respectively $n(\binom{d}{2} + 1)$, evaluations. For some models where a single evaluation takes a few minutes to one hour, this solution becomes rapidly intractable in case of large input space dimension.

Some improvements have been introduced by Saltelli (2002) to reduce the number of evaluations but with a cost still depending of the dimension d of the input space. A solution to this issue lies in the use of the replication method as explained in the next section.

3. Replication method

The replication method has been described for first-order Sobol' indices estimation by Mara *et al.* [4]. It has been combined with the use of randomized orthogonal array for closed second-order Sobol' indices estimation in Tissot and Prieur [5]. We propose hereafter a description of these methods and their generalization to independent groups of dependent variables.

3.1. Replication method for first-order Sobol' indices estimation

In their paper [4], Mara *et al.* propose a comparison between a deterministic sensitivity analysis and a stochastic sensitivity analysis. The deterministic approach is achieved using the improved FAST method. The stochastic methodology is performed with a modification of the Sobol' method using the concept of replication. This improved Sobol' method consists in two replicated Monte Carlo designs in the case of d independent inputs. As described below, this method can be extended for groups of independent inputs.

Let us consider p independent one-dimensional or multidimensional input variables $X_1, \ldots, X_I, \vec{X}_{I+1}, \ldots, \vec{X}_p$ as in (3). Thus $p = I + m$.

$$\text{Let} \qquad D = \{X^j = (X_1^j, \ldots, X_I^j, \vec{X}_{I+1}^j, \ldots, \vec{X}_p^j), 1 \leq j \leq n\}$$

be a design where X_i^j denotes the j-th sample of X_i. Each X^j is therefore a point in the d-dimensional input space. We say that a design D' is replicated from D if it is obtained through a column-wise permutation of D. More precisely it means that:

$$D' = \{X'^j = (X_1^{\pi_1(j)}, \ldots, X_I^{\pi_I(j)}, \vec{X}_{I+1}^{\pi_{I+1}(j)}, \ldots, \vec{X}_p^{\pi_p(j)}), 1 \leq j \leq n\}$$

where the vectors π_i $i \in \{1, \ldots, p\}$ are p independent random variables uniformly distributed in the set of permutations Π_n and $\pi_i(j) = \pi_i^j$ denotes the j-th component of π_i.

Let us denote by f the model and by Y_D and $Y_{D'}$ the two vectors of model outputs associated to D and D':

$$Y_D = \{Y_D^j = f(X_1^j, \ldots, X_I^j, \vec{X}_{I+1}^j, \ldots, \vec{X}_p^j), 1 \leq j \leq n\} \ ,$$

$$Y_{D'} = \{Y_{D'}^j = f(X_1^{\pi_1(j)}, \ldots, X_I^{\pi_I(j)}, \vec{X}_{I+1}^{\pi_{I+1}(j)}, \ldots, \vec{X}_p^{\pi_p(j)}), 1 \leq j \leq n\} \ .$$

To estimate the index S_k, $k \in \{1, \ldots, p\}$, the values of Y_D are rearranged with the corresponding permutation π_k. As a result, it looks like $Y_{D'}$ has

been constructed by varying all groups of inputs except the k-th. This is the same concept of "freezing" as the one introduced by Sobol' [1](see Appendix B for a detailed example).
Then using (4) with Y_D rearranged and $Y_{D'}$ in place of Y and Y_I we obtain $\widehat{S_k}$ (see Appendix B for an illustration of the method).

3.2. Replication method for closed second-order Sobol' indices estimation

In the case of closed second-order Sobol' indices estimation, the replication method does not consist anymore into a column-wise set of permutations. Indeed, to estimate closed second-order Sobol' indices, a way to "freeze" each subset of two variables has to be found. We need to find a structure allowing to get the same 2-sets of values in each subset of two columns of each replicated design. Such a structure has been introduced by Kishen [12] and further extended by Rao [13] and is known as orthogonal array. It is defined as follows:

Definition 1. A $t-(q,d,\lambda)$ orthogonal array ($t \leq d$) is a $\lambda q^t \times d$ matrix whose entries are chosen from a q-set of $\mathbb{N}$ such that in every subset of t columns of the array, every t-subset of points of this q-set appears in exactly λ rows.

From this definition by setting $t = 2$, we can construct a structure consisting of points in $\{1,\ldots,q\}^{\lambda q^2}$ where each 2-set of columns have the same 2-set of points λ times.

We present now a result from the method of differences introduced by Bose and Bush [14] to construct a $2-(q,p,1)$ linear orthogonal array. We begin with some useful definitions and a theorem resulting from the method of differences:

Definition 2. A linear orthogonal array is a $t-(q,d,\lambda)$ orthogonal array where $\lambda = 1$ and q is a prime number. The set of rows of a linear orthogonal array is a subspace of $(\mathbb{Z}/q\mathbb{Z})^d$.

Definition 3. Let G be the Galois field of order q, $GF(q)$, a $\lambda q \times d$ matrix $D(\lambda q, d, q)$ is a difference matrix (or difference scheme) of strength 2 if for every subset of 2 columns of the matrix, the vector of row-wise difference of the subset contains every elements of $GF(q)$ λ times.

Theorem 1. Let M be a $D(\lambda q, d, q)$ difference matrix of strength 2, B the vector of elements of $GF(q)$, $\oplus$ the Kronecker sum and T the transpose operator then $(M \oplus B)^T$ is a $2-(q, d, \lambda)$ orthogonal array.

Now, consider q a prime number and $G = GF(q)$ the Galois field associated, the multiplication table of G is a $D(q, q, q)$ difference matrix of strength 2 with $\lambda = 1$. From Theorem 1 we can construct a $2-(q, p, 1)$ linear orthogonal array: $A = (M \oplus B)^T$ where M is a sub-matrix of p columns of the multiplication table of G where p still denotes the number of groups of independent inputs. What is left is to connect this $2-(q, p, 1)$ orthogonal array to the replication method. We consider a $q \times p$ design:

$$D = \{X^j = (X_1^j, \ldots, X_I^j, \vec{X}_{I+1}^j, \ldots, \vec{X}_p^j), 1 \leq j \leq q\}$$

where $X_1, \ldots, X_I, \vec{X}_{I+1}, \ldots, \vec{X}_p$ are one-dimensional or multidimensional independent variables and q is a prime number. Denote by A the previous $2-(q, p, 1)$ orthogonal array. We can establish a direct bijection between the indices $\{1, \ldots, q\}$ of rows of D and the set $\{1, \ldots, q\}$ from which A is constructed. Each column of A will serve as new indices to build the respective column of the new design. Resulting from the construction procedure, in each column of A, each entry of $\{1, \ldots, q\}$ appears q times. This is due to the fact that each entry is involved in q different 2-sets. The number of rows of A is equal to q^2 meaning that using the previous bijection we can construct a design D_{OA} of q^2 rows where each column has the same entries of the corresponding column of D but re-indexed from the elements of the corresponding column of A:

$$D_{OA} = \{X^j = (X_1^{A_1^j}, \ldots, X_I^{A_I^j}, \vec{X}_{I+1}^{A_{I+1}^j}, \ldots, \vec{X}_p^{A_p^j}), 1 \leq j \leq q^2\}.$$

This gives us the first of the two needed replicated designs. To replicate D_{OA} we consider once again random vectors of permutations π_i in Π_q. We also have a direct bijection between the elements $\{1, \ldots, q\}$ of each π_i and the set $\{1, \ldots, q\}$ from which A is constructed. This allows the following construction of the second needed replicated design:

$$D'_{OA} = \{X'^j = (X_1^{\pi_1(A_1^j)}, \ldots, X_I^{\pi_I(A_I^j)}, \vec{X}_{I+1}^{\pi_{I+1}(A_{I+1}^j)}, \ldots, \vec{X}_p^{\pi_p(A_p^j)}), 1 \leq j \leq q^2\}$$

where each π_i is re-indexed from the elements of the corresponding column of A. In fact, the matrix whose columns are the vectors $\pi_1(A_1), \ldots, \pi_p(A_p)$ is also an orthogonal array (see Appendix A for details).

We obtain two designs of length q^2. Denote then $Y_{D_{OA}}$ and $Y_{D'_{OA}}$ the two vectors of model outputs associated to D_{OA} and D'_{OA}. To get the same "freezing" concept as in the last subsection, but this time for each subset of two coordinates $(k, l) \in \{1, \ldots, p\}^2$, we apply the following transformation:

$$\forall j \in \{1, \ldots, q^2\} : \; {Y^{A_k^j + A_l^j * q - q}} = {Y_{D_{OA}}}^j, \; {Y'}^{\pi_k(A_k^j) + \pi_l(A_l^j) * q - q} = {Y_{D'_{OA}}}^j .$$

As a result, Y and Y' have been constructed by varying all groups of inputs except the k-th and l-th. This allows us to estimate the index $\underline{S_{kl}}$ with the formula (4) by replacing Y_I with Y' (see Appendix C for a detailed example).

Remark 1. The construction of a $2 - (q, d, 1)$ linear orthogonal array is possible only if $q \geq d - 1$. The total number of evaluation required by the replication method will be $2q^2$.

Remark 2. For the sake of clarity, we have treated the case of first-order and closed second-order indices estimation separately. Actually, given that a $1 - (q, d, 1)$ orthogonal array is a matrix of permutations we could have written a general definition for both cases.

4. Sampling strategy under linear ordered constraints

In the previous section a method based on replicated designs to estimate Sobol' indices for independent groups of inputs was proposed. The fact that it handles groups of inputs is a way to tackle the problem of dependence within parameters. Indeed, each set of dependent inputs is treated as a group. The first- and second-order indices are calculated for the group, and not independently for each parameter (which would provide values that can not be interpreted easily). This method can be used for any kind of dependency, under the condition that one is able to properly sample from the input space; samples that are further needed to feed the designs. When inputs are independent, getting samples uniformly distributed in $[0, 1]$ is sufficient. When dependency is expressed through constraints, one has to create samples that satisfy the constraints, which is not an easy task in general. As mentioned in the introduction, the work presented in this paper is motivated by a variance-based sensitivity analysis we want to conduct on a land use and transport integrated model, called Tranus [15]. Such a model creates a numerical representation of the transport network on a territory, and calculates the spatial repartition of the economic activities. It is based on economic laws whose

parameters can be linked though ordered linear constraints. We thus focus on such constraints and propose an adapted sampling strategy based on the simplex geometric structure. This strategy is further enhanced with a space filling methodology.

4.1. Sampling strategy based on the simplex to handle constraints

We consider the same vector of inputs X as defined in (3). Each unidimensional variable is uniformly distributed in $[0,1]$. Each multidimensional variable $\vec{X}_{I+i},\ i \in \{1,\dots,m\}$ contains k_i variables valued in $[0,1]$ and is subject to the following linear ordered constraints:

$$X_{I+1+\Sigma_{i-1}} \leq X_{I+2+\Sigma_{i-1}} \leq \cdots \leq X_{I+\Sigma_i}\ . \tag{5}$$

In case of absence of constraints, the joint distribution of each multidimensional variable $\vec{X}_{I+i}$ is a uniform distribution on the unit k_i-hypercube. The introduction of constraints (5) transforms the support of the joint uniform distribution in a k_i-simplex. The general definition of a simplex is the following:

Definition 4. A simplex, or k-simplex, is the generalization of a triangle (2-simplex) or tetrahedron (3-simplex) to any dimension k. Let Δ_k be the simplex with vertices $P_0,\dots,P_k$ in $\mathbb{R}^k$:

$$\Delta_k = \{M = \sum_{j=0}^{k} w_j P_j \mid w_j \geq 0,\ \sum_{j=0}^{k} w_j = 1\}$$

To sample from a uniform distribution over a simplex we refer to the following theorem of L. Devroye [16]:

Theorem 2 (Devroye,1986). Let Δ_k be a simplex in $\mathbb{R}^k$ with vertices $P_0,\dots,P_k$ and $U_1,\dots,U_k$ be independent variables uniformly distributed in $[0,1]$. We define $W_0,\dots,W_k$ the associated differences as follows:

$$\forall j \in \{1,\dots,k\},\ W_j = V_{j+1} - V_j,\ V_0 = 0,\ V_{k+1} = 1$$

where V_j denote the j-th ordered statistics constructed from the set $(U_i)_i$:

$$\forall j \in \{1,\dots,k\},\ V_j = \min_{i=1,\dots,k} U_i \setminus V_{j-1}.$$

Then the random variable $Z = \sum_{j=0}^{k} W_j P_j$ is uniformly distributed in the simplex Δ_k.

To each set of constraints (5) is associated a k_i-simplex, defined as follows:

$$\Delta^*_{k_i} = \{M = \sum_{j=0}^{k_i} w_j e^*_j \mid w_j \geq 0,\ \sum_{j=0}^{k_i} w_j = 1\}$$

with vertices the ordered vectors of the standard basis of $\mathbb{R}^{k_i}$:

$$e^*_0 = (0, 0, \ldots, 0, 0), e^*_1 = (0, 0, \ldots, 0, 1), \ldots, e^*_{k_i} = (1, 1, \ldots, 1, 1)$$

Since the vertices of $\Delta^*_{k_i}$ satisfy ordered linear constraints (5), then every point sampled in $\Delta^*_{k_i}$ will also satisfy those constraints. Sampling in $\Delta^*_{k_i}$ using the strategy of Theorem 2 is equivalent to sampling uniformly under a set of k_i linear ordered constraints (5).

This method enables to sample each of the m multidimensional variables contained in the vector of inputs X. For the remaining independent one-dimensional variables we use uniform random numbers. Algorithm 1 summarizes the whole sampling strategy to get n samples:

Algorithm 1 Sampling n points from d inputs under m disjoint k_i-sets of linear ordered constraints

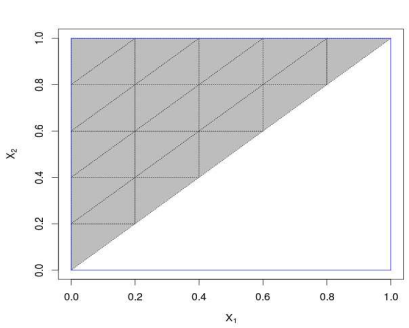
1. for each of the m multidimensional variables $\vec{X}_{I+i}$
   - create the unit ordered k_i-simplex
   - sample n points with Theorem 2
2. for each of the $d - \sum_{i=1}^{m} k_i$ one-dimensional variables sample n random uniform numbers.

4.2. Space filling strategy

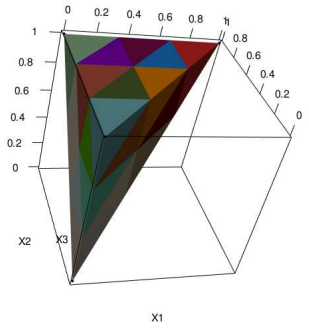
In the classical case of independent variables, one should prefer to use a Latin Hypercube rather than sampling each variable from a uniform distribution. Such a procedure ensures that the input space will be properly explored. To ensure the same property in our case of constrained input space, the proposed sampling method has to be improved as nothing prevents all the points to be located in a small area of the space. A way to do so is to

first subdivide each k_i-simplex into sub-simplices of order k_i and then sample from the latter. Algorithm 2 summarizes our subdivision method (see Appendix D for a detailed example). For the sake of clarity, in the following, k_i has been replaced by k. Figure 1 shows two examples of subdivisions for $k = 2$ and $k = 3$.

Figure 1: examples of subdivision of the unit ordered k-simplex



unit ordered 2-simplex with $n = 25$



unit ordered 3-simplex with $n = 27$

Let us remark that from Theorem 2, the uniform distribution in the unit ordered k-simplex in $\mathbb{R}^k$ corresponds to the joint distribution of the k ordered statistics. The l-th ordered statistic of a uniform distribution is a Beta variable : $X_{(l)} \sim B(l, k + 1 - l)$.

This remark can serve as a goodness of fit measure for our space filling sampling improvement of the uniform distribution over the unit ordered k-simplex. Indeed, since the marginal distributions are known, a Kolmogorov Smirnov test (ks.test) can be performed to compare our space filling strategy to a standard uniform sampling in the simplex. For each l-th ordered statistic, we perform the test for $k = 3$ and different sizes n of the sample. The test procedure is repeated $r = 100$ times. We take the average value of the r p-values obtained. Table 1 gives the averaged p-value obtained for each (n, l)-set. The general observation is that for each l-th ordered statistic the average p-value obtained for the space filling sampling is better than the one obtained for the standard sampling, whatever the number of samples is. This clearly shows the advantage of using the space filling strategy.

Algorithm 2 Space filling sampling of n points for a set of k linear ordered constraints

1. create the k^2 vertices of the unit k-hypercube and assign a number to each vertice according to the order of creation
2. identify the $k+1$ vertices whose coordinates satisfy the set of constraints
3. store those vertices into a $(k+1) \times k$ matrix M_{coor}, each column corresponding to one of the k coordinates
4. create the $k!$ vectors of permutations of Π_k
5. apply each permutation to the columns of M_{coor}, this creates a total of $k!$ matrices
6. for each one of the $k!$ matrices, match the vertices with those of the unit k-hypercube and retrieve the vector of matching numbers, this creates a total of $k!$ vectors of matching numbers that we note $\omega_j,\ j \in \{1, \ldots, k!\}$
7. choose $\alpha = \texttt{floor}(n^{\frac{1}{k}})$ (respectively $\alpha = n^{\frac{1}{k}}$ if $n^{\frac{1}{k}} \in \mathbb{N}$) the number of levels
8. subdivide the unit k-hypercube into $(\alpha+1)^k$ (respectively α^k if $n^{\frac{1}{k}} \in \mathbb{N}$) sub-hypercubes. For each sub-hypercube, assign a number to each vertices following the same order of numeration as in 1.
9. for each sub-hypercube:
   - for each ω_j, conserve the vertices whose indices correspond to ω_j, this gives $k!$ sets of vertices
   - among those sets keep only those whose coordinates satisfy the constraints
10. randomly discard $(\alpha+1)^k - n$ of all the sets created to keep only n sets
11. sample one point using each set in theorem 2 (each set are the vertices of a simplex) to get a total of n points

Table 1: Averaged p-values given by the ks.test to compare our space-filling sampling to the standard sampling on the unit ordered 3-simplex. The test is repeat $r = 100$ times for each l-th ordered statistic and for different value of number n of points sampled.

	standard sampling			space filling sampling		
n \ l	1	2	3	1	2	3
100	0.53	0.56	0.50	0.80	0.82	0.76
250	0.54	0.53	0.47	0.82	0.82	0.75
500	0.51	0.48	0.53	0.93	0.98	0.93
1250	0.49	0.51	0.52	0.93	0.96	0.93
2500	0.46	0.47	0.46	0.93	0.97	0.95
3750	0.46	0.52	0.50	0.96	0.98	0.96
5000	0.46	0.49	0.49	0.92	0.94	0.93

5. Summary of the whole procedure

Consider the vector of inputs

$$X = (X_1, \ldots, X_I, \underbrace{X_{I+1}, \ldots, X_{I+k_1}}_{\vec{X}_{I+1}}, \ldots, \underbrace{X_{I+1+\Sigma_{i-1}}, \ldots, X_{I+\Sigma_i}}_{\vec{X}_{I+i}},$$
$$\ldots, \underbrace{X_{I+k_1+k_2+\cdots+k_{m-1}+1}, \ldots, X_d}_{\vec{X}_{I+m}}) \text{ where } \Sigma_i = \sum_{l=0}^{i} k_l,\ k_0 = 0\ .$$

Each multidimensional variable $\vec{X}_{I+i},\ i \in \{1, \ldots, m\}$ contains k_i variables and is subject to the following linear ordered constraints:

$$X_{I+1+\Sigma_{i-1}} \leq X_{I+2+\Sigma_{i-1}} \leq \cdots \leq X_{I+\Sigma_i}\ .$$

Let d be the dimension of input space, $K = \sum_{i=1}^{m} k_i$ be the number of constrained inputs, m be the number of sets of constrained variables then the number of grouped Sobol' indices to estimate is $p = d - K + m$ for first-order indices and $\binom{p}{2}$ for closed second-order indices.

Let $(A_i^j)_{i=1..p, j=1..n}$ be an orthogonal array of strength t. $t = 1$ respectively 2 to estimate first- respectively second-order indices. Let $n = q^t \in \mathbb{N}^*$ be the number of samples of the input space with q a prime number if $t = 2$. Denote $(D_{OA_i}^j)_{i=1,\ldots,d,\ j=1,\ldots,n}$ and $(D'_{OA_i}{}^j)_{i=1,\ldots,d,\ j=1,\ldots,n}$ the two replicated designs.

1. For the independent inputs $X_i \;\; i \in \{1, \ldots, I\}$, construct the replicated designs sampling from Latin Hypercube [5] to get space filling uniform distribution.

$$\forall i \in \{1, \ldots, I\} : \qquad D_{OA}{}_{\;i}^{\;j} = \frac{A_i^j - U_i^{A_i^j}}{q},$$
$$D'_{OA}{}_{\;i}^{\;j} = \frac{\pi_i(A_i^j) - U_i^{\pi_i(A_i^j)}}{q},$$

where the π_i and the U_i^j are independent random variables uniformly distributed in Π_n and $[0, 1]$.

2. For the constrained inputs $\vec{X}_{I+i} \;\; i \in \{1, \ldots, m\}$, construct the replicated designs sampling from algorithm 2 to get a space filling uniform distribution in the corresponding ordered k_i-simplices.

$$\forall i \in \{1, \ldots, m\} : \quad D_{OA}{}_{\;I+1+\Sigma_{i-1}, I+2+\Sigma_{i-1}, \ldots, I+\Sigma_i}^{\;j} = \sum_{l=0}^{k_i} W_l^{A_i^j} P_l^{A_i^j},$$
$$D'_{OA}{}_{\;I+1+\Sigma_{i-1}, I+2+\Sigma_{i-1}, \ldots, I+\Sigma_i}^{\;j} = \sum_{l=0}^{k_i} W_l^{\pi_i(A_i^j)} P_l^{\pi_i(A_i^j)},$$

where $W_0^{A_i^j}, \ldots, W_{k_i}^{A_i^j}$ (respectively $W_0^{\pi_i(A_i^j)}, \ldots, W_{k_i}^{\pi_i(A_i^j)}$) are the associated differences constructed from $U_{I+1+\Sigma_{i-1}}^{A_i^j}, \ldots, U_{I+\Sigma_i}^{A_i^j}$ (respectively $U_{I+1+\Sigma_{i-1}}^{\pi_i(A_i^j)}, \ldots, U_{I+\Sigma_i}^{\pi_i(A_i^j)}$) and $P_0, \ldots, P_{k_i}$ are the columns of the $n \times (k_i+1)$ matrix P storing the sets of vertices resulting from Algorithm 2.

3. Denote then $Y_{D_{OA}}$ and $Y_{D'_{OA}}$ the two vectors of model outputs associated to D_{OA} and D'_{OA} we compute the Sobol' indices with the methodology presented in section 3.

Remark 3. As we stated at the beginning of section 4, we can apply the replication method to any kind of distributions under any type of correlation. The only requirement is to be able to generate a design or in other words to be able to sample each multidimensional variable (see section 6 for a multivariate Gaussian example).

6. Application to test functions

In this section the replication method is applied to standard test functions with sets of linear ordered constraints and to a case of a multivariate normal distribution. We compare the results to those given by the Saltelli method and the standard method of Sobol' that we both generalized to the case of groups of inputs (a first generalization of the standard method of Sobol' can be found in [8]). For both the standard method and our replication method we used the efficient estimator with formula given in (4). The size N of the design of experiments (DoE) is different for each method. Let n denotes the numbers of points in the d-dimensional input space:

- for the replication method we get $N = 2n$ (each of the two replicated designs has n points) for first-order indices and $N = 2n$ for closed second-order Sobol' indices where $\sqrt{n}$ has to be a prime number
- for Saltelli's method we get $N = n(p + 2)$ for first-order indices and $N = n(2p + 2)$ for closed second-order indices where p is the number of independent one-dimensional or multidimensional input variables
- for the standard method we get $N = n(p + 1)$ for first-order indices and $N = n(\binom{p}{2} + 1)$ for closed second-order indices

From the user point of view, the total number of evaluations required is one of the most important aspect of the method. Thus, the comparison are made for similar computational time. For each method, we chose n to get the same value N.

6.1. Sobol's g-function

We consider the g-function introduced by Sobol' [1]. The g-function is defined as follows:

$$f(X_1, \ldots, X_d) = \prod_{i=1}^{d} g_i(X_i),$$

where

$$g_i(X_i) = \frac{|4X_i - 2| + a_i}{1 + a_i}, \ a_i \geq 0.$$

Each value a_i determines the relative importance of the X_i. When the value of a_i gets closer to zero the variable X_i becomes more influent. We choose here $d = 4$ and $(a_1 = 0, a_2 = 1, a_3 = 3, a_4 = 6)$.

We consider the following sets of characteristics:

- $X_3 \leq X_4$ with (X_3, X_4) uniformly distributed in Δ_2^*
- X_1 and X_2 are independent and both uniformly distributed in $[0, 1]$.

Since f has an analytical expression, Sobol' indices can be calculated by symbolic integrals to evaluate their theoretical value. Each of the three methods are used to compute the three first-order Sobol' indices and the three closed second-order Sobol' indices. The estimation procedure is repeated $r = 100$ times to get a sample of estimated indices; and this for various values of N.

To evaluate the precision of each method, are shown for each index:

- boxplots drawn depending on increasing values of N ;
- evolutions of the error relative to the true value, depending on N. The error calculated is a variation of the SAE (Sum of Absolute Error) indicator used in [4], and is defined as:

$$SAE_N = \sum_{k=1}^{r} |\underline{\hat{S}}_{k,N} - S|,$$

  where $\underline{\hat{S}}_{k,N}$ is the k-th estimation of the index, and S the theoretical value of the index.

Results on first-order indices are shown on figure 2, while results on closed second-order Sobol' indices are shown on figure 3. In both cases, the first row corresponds to boxplots and the second row to error curves. Note that for all illustrations, black is associated to our replication method, blue to Saltelli approach and green to Sobol' method (also referred as standard method). The theoretical values of indices are indicated by red horizontal broken lines.

For the calculation of first-order indices, the observation is that the replication method gives better results, and particularly for low values of N. This observation is explained by the independence between the computational cost of our approach and the input space dimension. Thus, this observation is emphasized when we increase the number p of independent groups of inputs. The Saltelli's method gives better results when it comes to estimate indices with low value rather than indices with high value. At the opposite both the standard method and the replication method are better at estimating indices with high value but this could be due to the choice of the estimator rather

than from the method itself (see Owen [17] for a discussion on the estimation of small indices).

The replication method performs also better compared to the other two approaches for the calculation of closed second-order indices. The Saltelli method is once again better at estimating indices with low value and the other two methods are better at estimating indices with high value. Despite the difference, for low value indices our method still stays competitive.

In order to test our method facing a high number of inputs, we consider the Sobol' g-function with a dimension of input space now equals to 50. The following characteristics are considered:

- (X_1, X_{50}), (X_2, X_{49}), (X_3, X_{48}), (X_4, X_{47}), (X_5, X_{46}) uniformly distributed in Δ_2^*,
- (X_6, X_{10}, X_{40}), (X_7, X_{11}, X_{39}), (X_8, X_{12}, X_{38}), (X_9, X_{13}, X_{37}), (X_{19}, X_{30}, X_{31}) uniformly distributed in Δ_3^*,
- $(X_{14}, X_{20}, X_{21}, X_{36})$, $(X_{15}, X_{22}, X_{23}, X_{35})$, $(X_{16}, X_{24}, X_{25}, X_{34})$, $(X_{17}, X_{26}, X_{27}, X_{33})$, $(X_{18}, X_{28}, X_{29}, X_{32})$ uniformly distributed in Δ_4^*,
- $(X_{41}, X_{42}, X_{43}, X_{44}, X_{45})$ uniformly distributed in Δ_5^*.

The values of the a_i are the following: $(a_1 = ... = a_{10} = 0, a_{11} = ... = a_{20} = 1, a_{21} = ... = a_{30} = 2, a_{31} = ... = a_{40} = 4, a_{41} = ... = a_{50} = 6)$.

Since there is a substantial number of indices, we choose not to draw all boxplots but instead to compute the two errors. The average of the p SAE_N calculated for each first-order index is referred as $ASAE_N^1$. The average of the $\binom{p}{2}$ SAE_N calculated for each closed second-order index is referred as $ASAE_N^2$.

Table 2 gives the values of $ASAE_N^1$ for each method and each value of N. Figure 4 right represents the evolution curves of the $ASAE_N^2$. Since there is 120 closed second-order indices to estimate, the difference in that case between the results of the three methods is even more noticeable. Figure 4 left represents the evolution curves of the $ASAE_N^1$ given N for the three methods. It clearly illustrates the benefit of the independence of the replication method's total cost to the dimension of the input space. Even for $N = 10^4$ the replication method is still in average two times better than the other two methods.

Figure 2: Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions. The color black is for the replication method, the blue for the Saltelli method and the green for the standard method. At the top: boxplot representation for different values of N. At the bottom: curve of the SAE_N for different values of N.

Sobol' index S_1 | Sobol' index S_2 | Sobol' index $S_{3,4}$

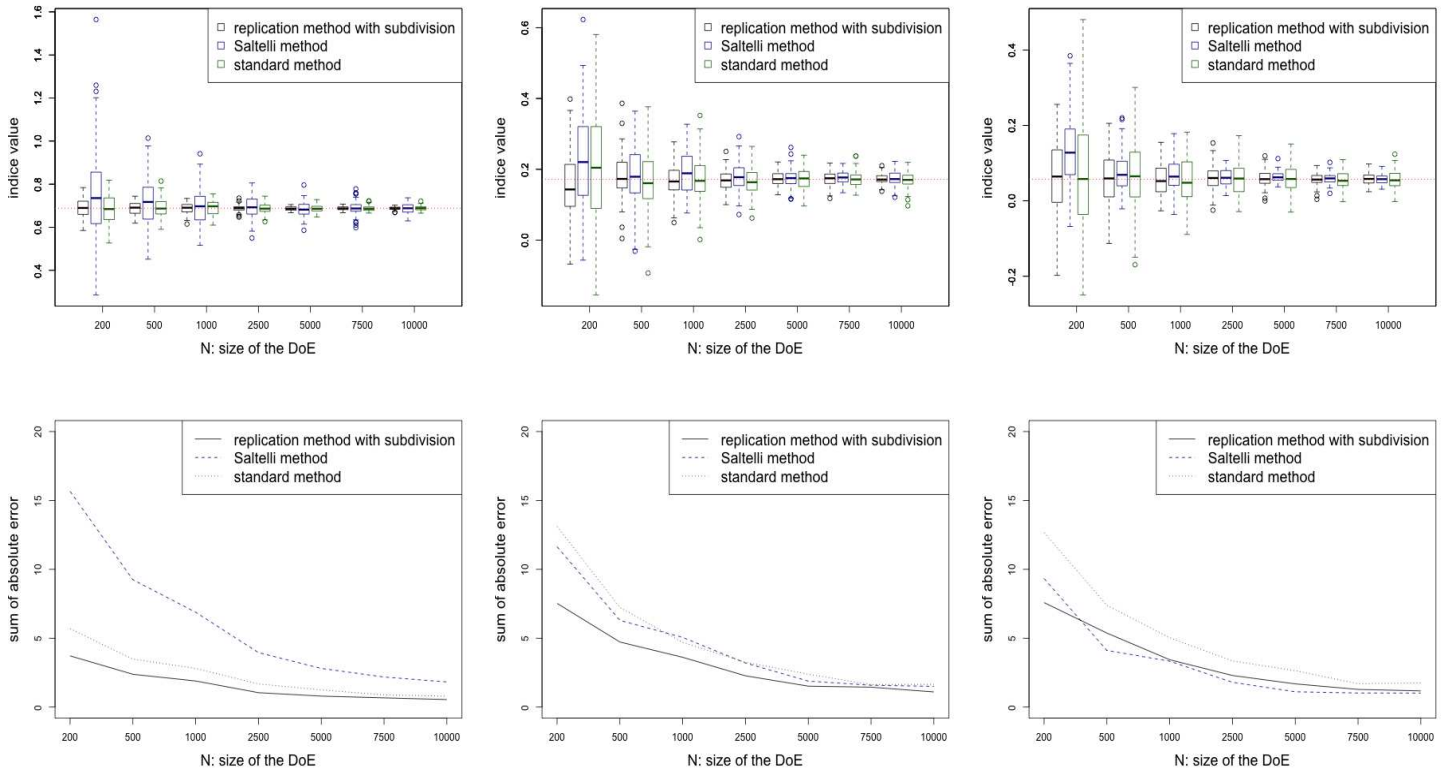


Figure 3: Estimation of closed second-order Sobol' indices given by the three methods for $r = 100$ repetitions. The color black is for the replication method, the blue for the Saltelli method and the green for the standard method. At the top: boxplot representation for different values of N. At the bottom: curve of the SAE_N for different values of N.

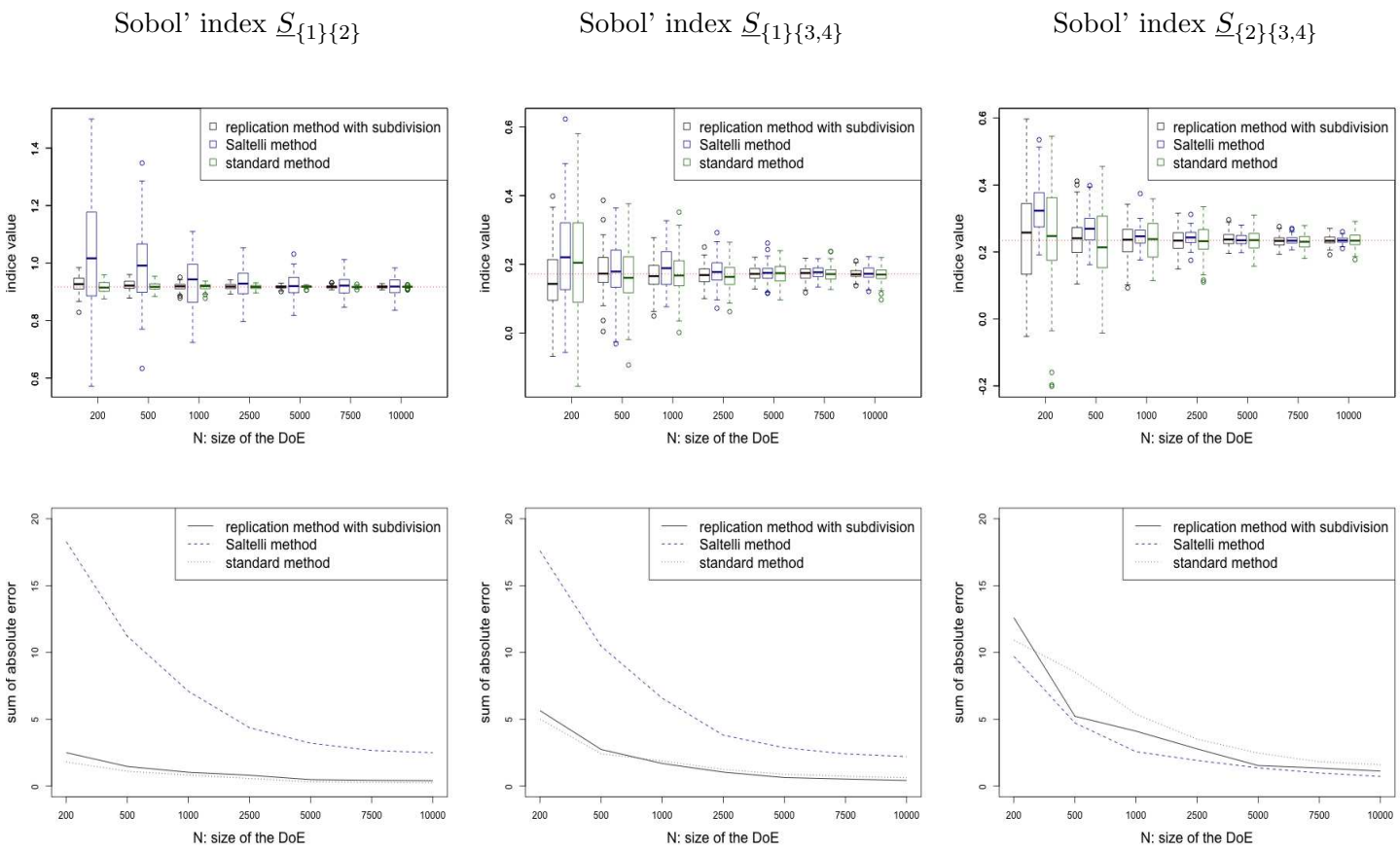
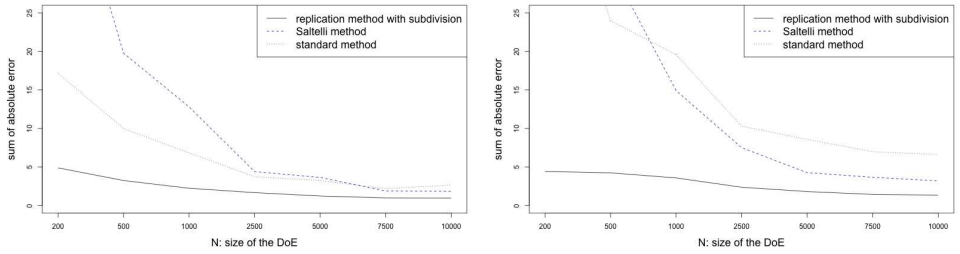


Table 2: Values of $ASAE_N^1$ for each method and for different size N of the design of experiments

size N	replicated method	Saltelli method	standard method
200	4.80	54.39	16.62
500	3.40	39.27	7.40
1000	2.35	16.59	5.80
2500	1.61	4.18	3.35
5000	1.18	3.41	2.97
7500	1.01	2.12	2.54
10^4	0.93	2.02	1.78

Figure 4: Curves of $ASAE_N^1$ (left figure) and $ASAE_N^2$ (right figure) for each method depending on N



6.2. Bratley et al. Function

For the second example, we consider the function introduced by Bratley *et al.* The function is defined as follows:

$$f(X_1, \ldots, X_d) = \sum_{i=1}^{d} (-1)^i \prod_{j=1}^{i} X_j \ .$$

We consider the following characteristics:

- $X_3 \leq X_4$ with (X_3, X_4) uniformly distributed in Δ_2^*
- X_1 and X_2 are independent and both uniformly distributed in $[0, 1]$.

We use the same estimation procedure as in the last subsection and we draw the same graphical results (Figure 5 and Figure 6). Conclusions are similar to the ones for the g-function of Sobol'. It shows the efficiency of our approach with respect to more standard approaches such as Saltelli or Sobol'.

6.3. Correlated multivariate normal distribution

As stated in Remark 3, this example of a correlated multivariate normal distribution is to shows that the replication method can be used in a general case of correlated inputs at the condition that we know how to sample each multidimensional input. Furthermore, the case of multivariate normal distribution with correlation terms is a common example of dependence for the inputs.

Let Σ be a positive-definite matrix, non necessarily diagonal. As in the previous subsection we write $X = (X_1, \ldots, X_I, \vec{X}_{I+1}, \ldots, \vec{X}_{I+m})$ where the multidimensional variables $\vec{X}_{I+j}$, $j \in \{1, \ldots, m\}$ are independent random gaussian vectors with mean vector μ and covariance matrix Σ. Based on section 2.2 we can estimate the Sobol' indices for the groups of correlated variables $\vec{X}_{I+i}$.

To sample such a distribution we first use a Cholesky decomposition of the covariance matrix $\Sigma = LL^T$. Then, let $Z = (Z_0, \ldots, Z_d)$ be a vector whose components are independent and follow a standard normal distribution. We obtain the true sampling from the formula: $X = \mu + LZ$. We obtain the two replicated designs of experiments D_{OA} and D'_{OA} with the formula from section 3 applied to our previous sample of X. As an example we consider the following function inspired from the function presented in [18]:

$$f(X_1, X_2, X_3, X_4) = g_1(X_1, X_2) + g_2(X_2) + g_3(X_3) + g_4(X_4)$$

$$\text{where } \begin{cases} g_1(X_1, X_2) = (2(X_1 - \mu_1) + 1)(3(X_2 - \mu_2) + 2) \\ g_2(X_2) = 2(X_2 - \mu_2)^2 + X_2 - \mu_2 + 3 \\ g_3(X_3) = 1 + 2(X_3 - \mu_3) + 2(X_3 - \mu_3)^2 + 3(X_3 - \mu_3)^3 \\ g_4(X_4) = 1 + 4(X_4 - \mu_4) \end{cases}$$

and $X = (X_1, X_2, X_3, X_4)$ follows a centered multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ with parameters:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & 0 & 0 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix} \quad \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .$$

Figure 5: Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions. The color black is for the replication method, the blue for the Saltelli method and the green for the standard method. At the top: boxplot representation for different values of N. At the bottom: curve of the SAE_N for different values of N.

Sobol' index S_1 | Sobol' index S_2 | Sobol' index $S_{3,4}$

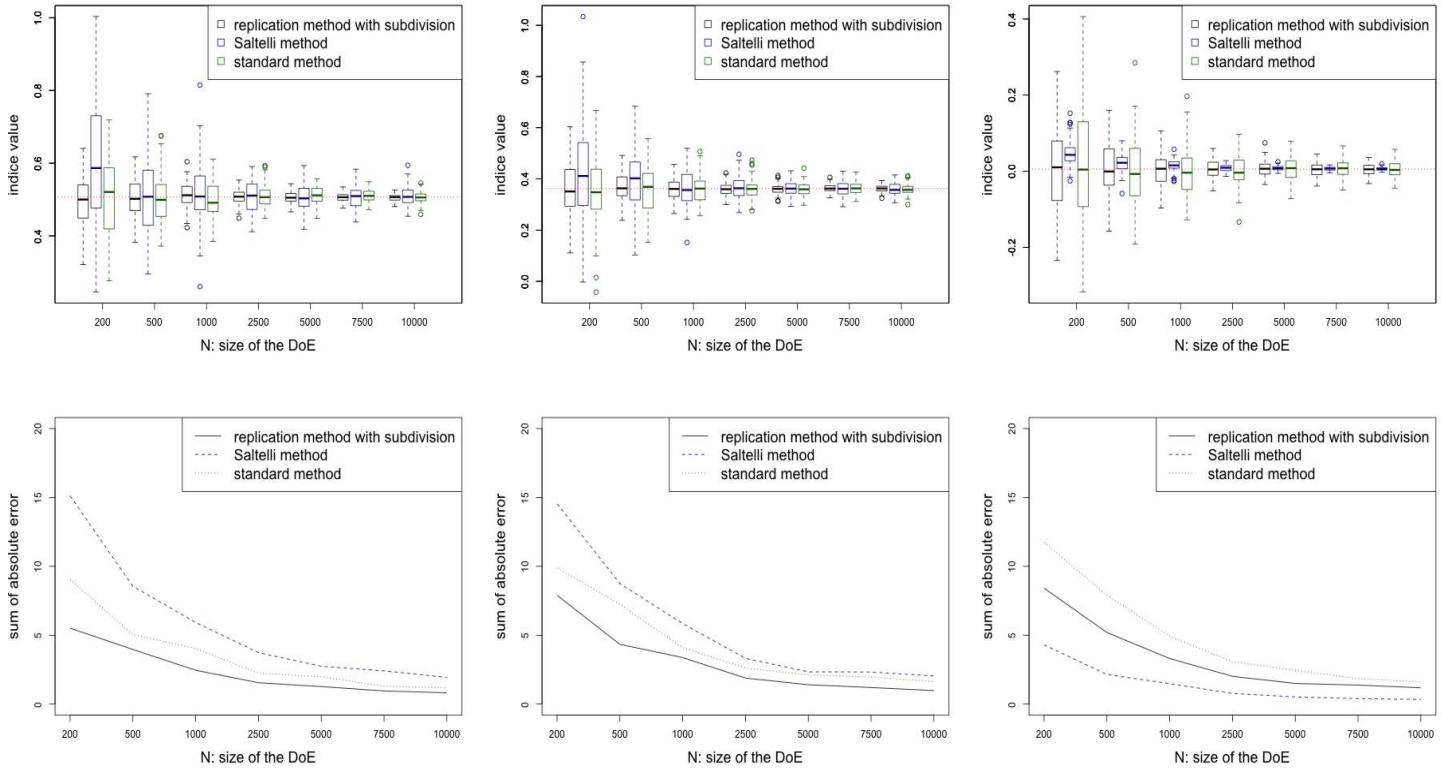
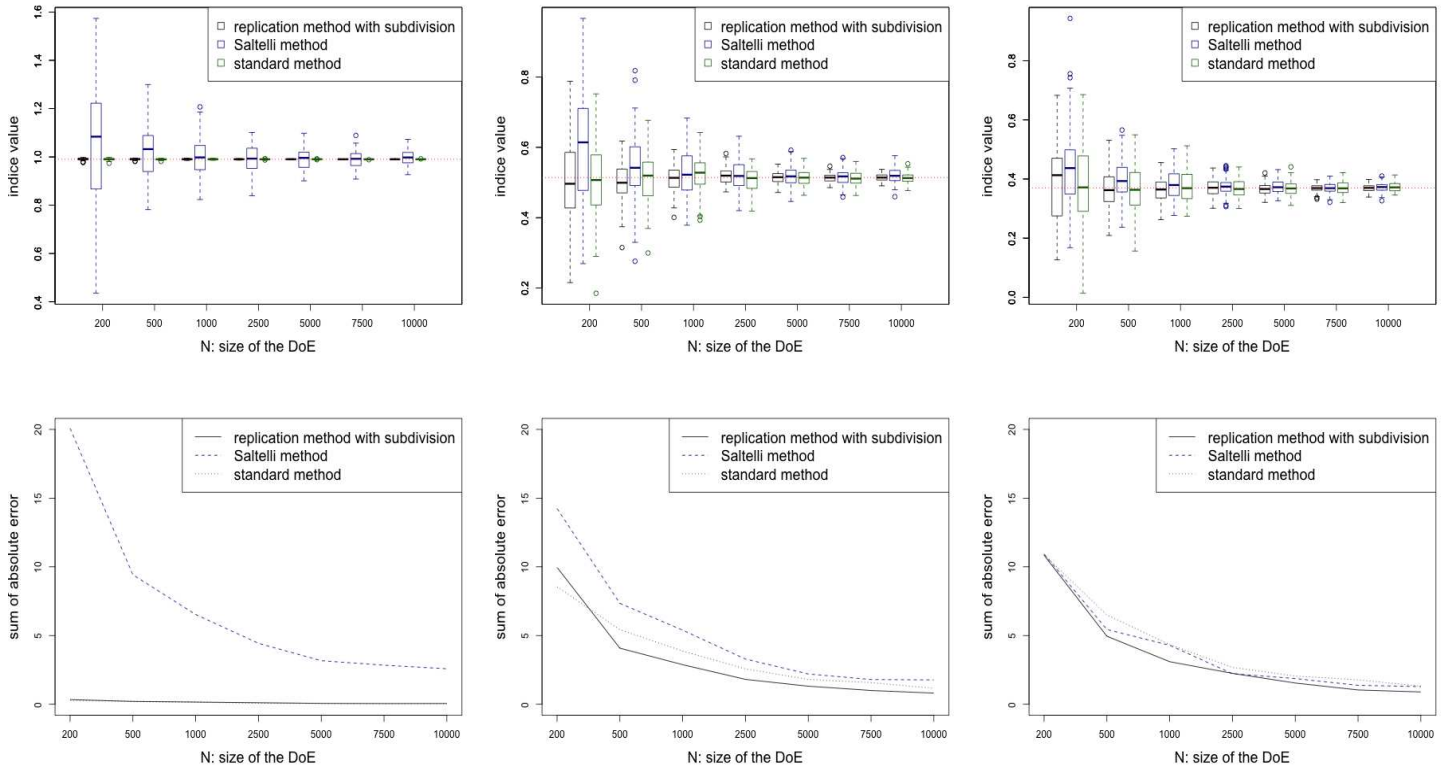


Figure 6: Estimation of closed second-order Sobol' indices given by the three methods for $r = 100$ repetitions. The color black is for the replication method, the blue for the Saltelli method and the green for the standard method. At the top: boxplot representation for different values of N. At the bottom: curve of the SAE_N for different values of N.

Sobol' index $\underline{S}_{\{1\}\{2\}}$ Sobol' index $\underline{S}_{\{1\}\{3,4\}}$ Sobol' index $\underline{S}_{\{2\}\{3,4\}}$



The first two variables X_1,X_2 are correlated and we chose the following values for the terms in Σ:

$$\sigma_1 = \sigma_2 = 0.3,\ \sigma_3 = 0.5,\ \sigma_4 = 0.6,\ \rho_{12} = 0.2$$

We use the same estimation procedure as in the last two subsections and we draw the results only for first-order Sobol' indices since the sum of the true values of those indices is equal to 0.999 which means that interactions are negligible.

Looking at the results in Figure 7 our replication method still gives good results even in this case where there is no space filling strategy involved. The results given by the Saltelli method are in accordance with the results from the previous two subsections.

Conclusion

In this paper we proposed a methodology to estimate first-order or closed second-order grouped Sobol' indices using replicated designs to handle groups of dependent inputs. In our application on the LUTI model of Tranus this dependency translates into groups of linear ordered inputs. Thus, through an algorithm (Algorithm 2) we have presented a space filling strategy based on the simplex structure to sample points satisfying those constraints and a detailed procedure to estimate the associated Sobol' indices. Compared to the standard method of Sobol' [1] or to Saltelli's approach [2], our new estimation procedure has a better precision for a given cost, especially in the case of high number of inputs. An application on the Tranus LUTI model will be the subject of a future work via a grid deployment.

References

[1] I. M. Sobol', Sensitivity indices for nonlinear mathematical models, Mathematical Modeling and Computational Experiment 1 (1993) 407–414.

[2] A. Saltelli, Making best use of model evaluations to compute sensitivity indices, Computer Physics Communications 145 (2002) 280–297.

[3] M. D. McKay, Evaluating prediction uncertainty, Technical Report, Los Alamos National Laboratory Report NUREG/CR- 6311, LA-12915-MS., 1995.

[4] T. A. Mara, O. R. Joseph, Comparison of some efficient methods to evaluate the main effect of computer model factors, Journal of Statistical Computation and Simulation 78 (2008) 167–178.

[5] J. Y. Tissot, C. Prieur, Estimating Sobol's indices combining Monte Carlo estimators and Latin hypercube sampling, Technical Report, 2014. URL: http://hal.archives-ouvertes.fr/hal-00743964.

[6] C. Lemieux, Monte Carlo and quasi-Monte Carlo sampling, Springer Series in Statistics, 2009.

[7] A. Owen, Orthogonal arrays for computer experiments integration and visualization, Statistica Sinica 2 (1992) 280–297.

[8] J. Jacques, C. Lavergne, N. Devictor, Sensitivity analysis in presence of model uncertainty and correlated inputs, Reliability Engineering & System Safety 91 (2006) 1126–1134.

[9] W. Hoeffding, A class of statistics with asymptotically normal distributions, Annals of Mathematical Statistics 19 (1948) 293–325.

[10] A. Janon, T. Klein, A. Lagnoux, M. Nodet, C. Prieur, Asymptotic normality and efficiency of two Sobol' index estimators, Technical Report, ESAIM P&S. Online publication June 06 2013. URL: http://dx.doi.org/10.1051/ps/2013040.

[11] H. Monod, C. Naud, D. Makowski, Uncertainty and sensitivity analysis for crop models, Elsevier, 2006, pp. 55–100.

[12] K. Kishen, On latin and hyper-graeco cubes and hypercubes, Current Science 11 (1942) 98–99.

[13] C. R. Rao, Hypercubes of strength "d" leading to confounded designs in factorial experiments, Bulletin of the Calcutta Mathematical Society 38 (1946) 67–78.

[14] R. C. Bose, K. A. Bush, Orthogonal arrays of strength two and three, The Annals of Mathematical Statistics 23 (1952) 508–524.

[15] T. Barra, Mathematical description of TRANUS, Technical Report, Modelistica, 1999. URL: http://www.tranus.com/tranus-english.

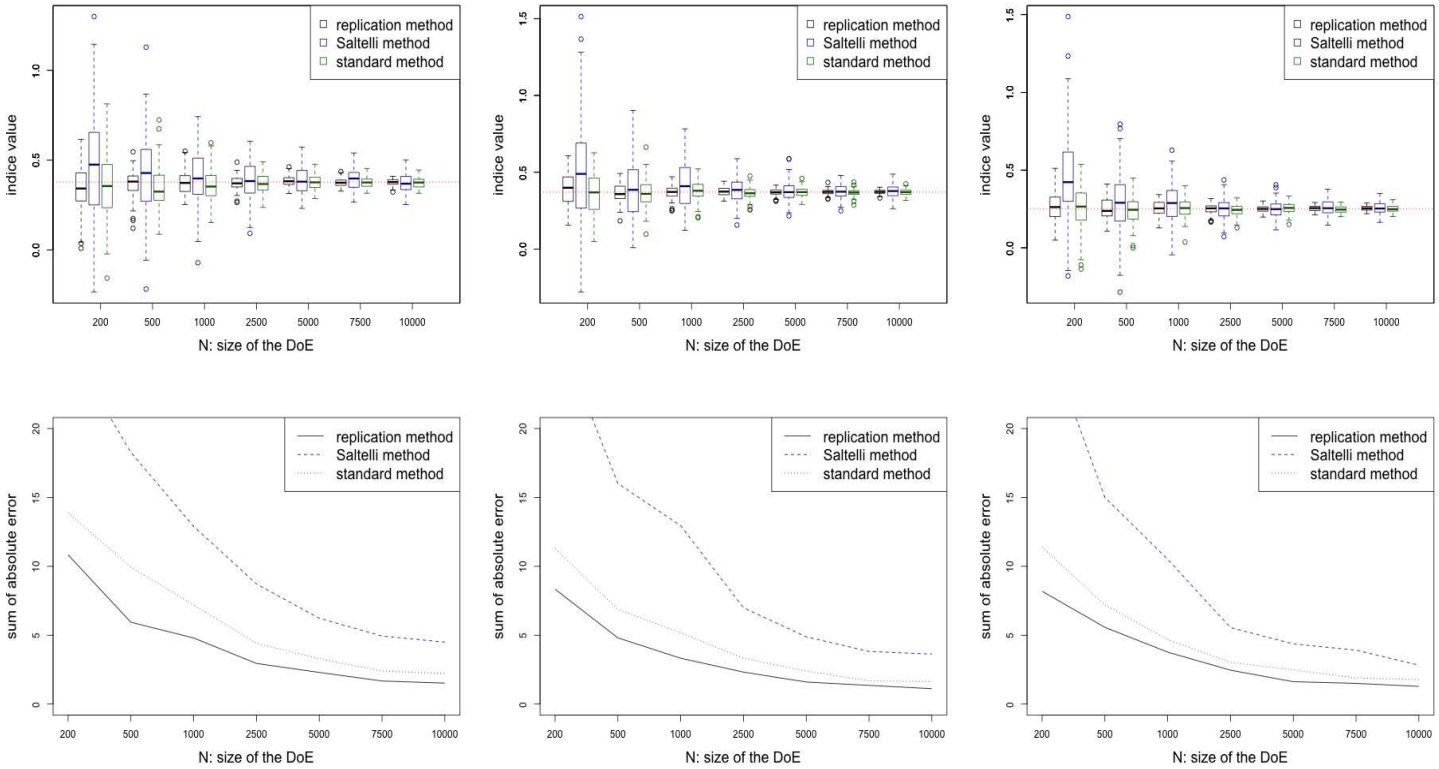
[16] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, 1986.

[17] A. Owen, Better estimation of small sobol' sensitivity indices, ACM Trans. Model. Comput. Simul. 23 (2013) 11:1–11:17.

[18] G. Li, H. Rabitz, General formulation of hdmr component functions with independent and correlated variables, Journal of Mathematical Chemistry 50 (2011) 99–130.

Figure 7: Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions. The color black is for the replication method, the blue for the Saltelli method and the green for the standard method. At the top: boxplot representation for different values of N. At the bottom: curve of the SAE_N for different values of N.

Sobol' index S_3 Sobol' index S_4 Sobol' index $S_{1,2}$



Appendix A. Proofs

Proposition 1. Let A be a 2-$(q, p, 1)$ linear orthogonal array of elements in $\{1, \ldots, q\}$. Let $\pi_1(A_1), \ldots, \pi_p(A_p) \in \Pi_q$, p independent vectors of permutations. Then the matrix A_π whose columns are the vectors $\pi_1(A_1), \ldots, \pi_p(A_p)$ is also a 2-$(q, p, 1)$ orthogonal array.

Proof. From Definition 1, we need to demonstrate that for every subset of 2 columns of A_π, every 2-set appears exactly one time. This would mean that every subset of 2 columns of A_π is identical to the space $\{1, \ldots, q\} \times \{1, \ldots, q\}$. Let us consider only A_1, A_2 and π_1, π_2, the demonstration will be identical for the other subset of 2 columns. Consider the following application:

$$f\colon \left\{ \begin{array}{ccc} \{1, \ldots, q\} \times \{1, \ldots, q\} & \to & \{1, \ldots, q\} \times \{1, \ldots, q\} \\ (x, y) & \mapsto & (\pi_1(x), \pi_2(y)) \end{array} \right.$$

f is a bijection since π_1, π_2 are two bijections from $\{1, \ldots, q\}$ to $\{1, \ldots, q\}$. Now if x are elements of A_1 and y are elements of A_2 this means that the subset $(\pi_1(A_1)\ \pi_2(A_2))$ have the same 2-set of $(A_1\ A_2)$. Applying this to every other subset of 2 columns leads to the result.

Proposition 2. We want to prove that the following transformation allows us to freeze each set of two coordinates $(k, l) \in \{1, \ldots, p\}^2$ allowing us to estimate each closed second-order index S_{kl}:

$$\forall j \in \{1, \ldots, q^2\} :\ {Y^{A_k^j + A_l^j * q - q}}^j = {Y_{D_{OA}}}^j,\ {Y'^{\pi_k(A_k^j) + \pi_l(A_l^j) * q - q}}^j = {Y_{D'_{OA}}}^j$$

Proof. To simplify the proof consider $k = 1$, $l = 2$, the method will be the same for the other set of two coordinates. Consider the following application:

$$f\colon \left\{ \begin{array}{ccc} \{1, \ldots, q\} \times \{1, \ldots, q\} & \to & \{1, \ldots, q^2\} \\ (x, y) & \mapsto & x + y * q - q \end{array} \right.$$

f is a bijection. To prove it, we only have to demonstrate that f is injective. Consider (x, y) and (x', y') in $\{1, \ldots, q\} \times \{1, \ldots, q\}$ such that:

$$x - x' = (y' - y) * q$$

this implies

$$x - x' \equiv 0[q]$$

given that

$$(x, x') \in \{1, \ldots, q\} \times \{1, \ldots, q\}$$

we obtain

$$x = x'.$$

Then

$$(y' - y) * q = 0$$

implies

$$y = y'.$$

Since A_1^j and $A_2^j \in \{1, \ldots, q\}$ (resp. $\pi_1(A_1^j)$ and $\pi_2(A_2^j)$), this means that every element of $\{1, \ldots, q^2\}$ is obtained from a unique 2-set of $(A_1\ A_2)$ (resp. $(\pi_1(A_1)\ \pi_2(A_2))$ thus each value of the vectors Y and Y' are calculated from the same set of the first two coordinates. We get the same 'freezing' result for every other set of two coordinates.

Appendix B. Example of the replication method for first-order indices

$$D = X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix} \end{array} \qquad \Pi = \begin{array}{c} \begin{array}{cccc} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{array} \\ \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} \end{array}$$

construction of D':

$$\Pi = \begin{pmatrix} \boxed{2} & \boxed{1} & \boxed{3} & \boxed{1} \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} \longrightarrow X = \begin{pmatrix} 11 & \boxed{21} & 31 & \boxed{41} \\ \boxed{12} & 22 & 32 & 42 \\ 13 & 23 & \boxed{33} & 43 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ \boxed{3} & \boxed{3} & \boxed{1} & \boxed{2} \\ 1 & 2 & 2 & 3 \end{pmatrix} \longrightarrow X = \begin{pmatrix} 11 & 21 & \boxed{31} & 41 \\ 12 & 22 & 32 & \boxed{42} \\ \boxed{13} & \boxed{23} & 33 & 43 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix} \longrightarrow X = \begin{pmatrix} \boxed{11} & 21 & 31 & 41 \\ 12 & \boxed{22} & \boxed{32} & 42 \\ 13 & 23 & 33 & \boxed{43} \end{pmatrix}$$

resulting in:

$$D' = \begin{pmatrix} 12 & 21 & 33 & 41 \\ 13 & 23 & 31 & 42 \\ 11 & 22 & 32 & 43 \end{pmatrix}$$

for a model f, the associated response are:

$$Y_D = \begin{pmatrix} f(11,21,31,41) \\ f(12,22,32,42) \\ f(13,23,33,43) \end{pmatrix} \quad Y_{D'} = \begin{pmatrix} f(12,21,33,41) \\ f(13,23,31,42) \\ f(11,22,32,43) \end{pmatrix}$$

to estimate the index S_1 we re-sample Y_D with π_1:

$$\pi_1 = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \longrightarrow Y_{Dnew} = \begin{pmatrix} f(12,22,32,42) \\ f(13,23,33,43) \\ f(11,21,31,41) \end{pmatrix}$$

looking at both Y_{Dnew} and $Y_{D'}$, X_1 has been "frozen".

Appendix C. Example of the replication method for closed second-order indices

$$D = X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix} \end{array} \quad \Pi = \begin{array}{c} \begin{array}{cccc} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{array} \\ \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} \end{array}$$

construction of a $2-(3,4,1)$ orthogonal array $A = (M \oplus B)^T$:

$$A = \left(\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right)^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \end{pmatrix}^T$$

we can always add another column corresponding to the vector of elements

of $GF(3)$ repeated 3 times:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 2 \\ 2 & 2 & 2 & 0 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 \end{pmatrix} \iff A = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$

construction of D_{OA}:

$$A = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow X = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix}$$

we iterate the procedure over the rows of A, resulting in:

$$D_{OA} = \begin{pmatrix} 13 & 23 & 33 & 43 \\ 13 & 21 & 32 & 41 \\ 13 & 22 & 31 & 42 \\ 11 & 21 & 31 & 43 \\ 11 & 22 & 33 & 41 \\ 11 & 23 & 32 & 42 \\ 12 & 22 & 32 & 43 \\ 12 & 23 & 31 & 41 \\ 12 & 21 & 33 & 42 \end{pmatrix}$$

construction of D'_{OA}:

$$A = \begin{pmatrix} \boxed{3} & \boxed{3} & \boxed{3} & \boxed{3} \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix}$$

then:

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix} \longrightarrow X = \begin{pmatrix} \boxed{11} & 21 & 31 & 41 \\ 12 & \boxed{22} & \boxed{32} & 42 \\ 13 & 23 & 33 & \boxed{43} \end{pmatrix}$$

we iterate the procedure over the rows of A, resulting in:

$$D'_{OA} = \begin{pmatrix} 11 & 22 & 32 & 43 \\ 11 & 21 & 31 & 41 \\ 11 & 23 & 33 & 42 \\ 12 & 21 & 33 & 43 \\ 12 & 23 & 32 & 41 \\ 12 & 22 & 31 & 42 \\ 13 & 23 & 31 & 43 \\ 13 & 22 & 33 & 41 \\ 13 & 21 & 32 & 42 \end{pmatrix}$$

for a model f, the associated response are:

$$Y_{D_{OA}} = \begin{pmatrix} f(13,23,33,43) \\ f(13,21,32,41) \\ f(13,22,31,42) \\ f(11,21,31,43) \\ f(11,22,33,41) \\ f(11,23,32,42) \\ f(12,22,32,43) \\ f(12,23,31,41) \\ f(12,21,33,42) \end{pmatrix} \quad Y_{D'_{OA}} = \begin{pmatrix} f(11,22,32,43) \\ f(11,21,31,41) \\ f(11,23,33,42) \\ f(12,21,33,43) \\ f(12,23,32,41) \\ f(12,22,31,42) \\ f(13,23,31,43) \\ f(13,22,33,41) \\ f(13,21,32,42) \end{pmatrix}$$

to estimate the index S_{12} we re-sample $Y_{D_{OA}}$ the following way: denotes A_1 and A_2 the first two columns of A:

$$A_1 + A_2 * 3 - 3 = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \\ 2 \\ 1 \\ 2 \\ 3 \\ 2 \\ 3 \\ 1 \end{pmatrix} * 3 - 3 = \begin{pmatrix} 9 \\ 3 \\ 6 \\ 1 \\ 4 \\ 7 \\ 5 \\ 8 \\ 2 \end{pmatrix}$$

we construct Y with this new indexation:

$$\begin{cases} Y[9] = Y_{D_{OA}}[1] = f(13, 23, 33, 43) \\ Y[3] = Y_{D_{OA}}[2] = f(13, 21, 32, 41) \\ Y[6] = Y_{D_{OA}}[3] = f(13, 22, 31, 42) \\ Y[1] = Y_{D_{OA}}[4] = f(11, 21, 31, 43) \\ Y[4] = Y_{D_{OA}}[5] = f(11, 22, 33, 41) \\ Y[7] = Y_{D_{OA}}[6] = f(11, 23, 32, 42) \\ Y[5] = Y_{D_{OA}}[7] = f(12, 22, 32, 43) \\ Y[8] = Y_{D_{OA}}[8] = f(12, 23, 31, 41) \\ Y[2] = Y_{D_{OA}}[9] = f(12, 21, 33, 42) \end{cases}$$

Next, we re-sample $Y_{D'_{OA}}$ the following way:
Denotes A_π the matrix A re-indexed by Π:

$$A = \begin{pmatrix} \boxed{3} & \boxed{3} & \boxed{3} & \boxed{3} \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix}$$

then:

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix} \longrightarrow A_\pi = \begin{pmatrix} 1 & 2 & 2 & 3 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

we calculate the following indexation:

$$A_{\pi 1} + A_{\pi 2} * 3 - 3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 3 \\ 1 \\ 3 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix} * 3 - 3 = \begin{pmatrix} 4 \\ 1 \\ 7 \\ 2 \\ 8 \\ 5 \\ 9 \\ 6 \\ 3 \end{pmatrix}$$

we construct Y' with this new index:

$$\begin{cases} Y'[4] = Y_{D'_{OA}}[1] = f(11, 22, 32, 43) \\ Y'[1] = Y_{D'_{OA}}[2] = f(11, 21, 31, 41) \\ Y'[7] = Y_{D'_{OA}}[3] = f(11, 23, 33, 42) \\ Y'[2] = Y_{D'_{OA}}[4] = f(12, 21, 33, 43) \\ Y'[8] = Y_{D'_{OA}}[5] = f(12, 23, 32, 41) \\ Y'[5] = Y_{D'_{OA}}[6] = f(12, 22, 31, 42) \\ Y'[9] = Y_{D'_{OA}}[7] = f(13, 23, 31, 43) \\ Y'[6] = Y_{D'_{OA}}[8] = f(13, 22, 33, 41) \\ Y'[3] = Y_{D'_{OA}}[9] = f(13, 21, 32, 42) \end{cases}$$

looking at both Y and Y', the two inputs X_1, X_2 have been "frozen":

$$Y = \begin{pmatrix} f(11,21,31,43) \\ f(12,21,33,42) \\ f(13,21,32,41) \\ f(11,22,33,41) \\ f(12,22,32,43) \\ f(13,22,31,42) \\ f(11,23,32,42) \\ f(12,23,31,41) \\ f(13,23,33,43) \end{pmatrix} \quad Y' = \begin{pmatrix} f(11,21,31,41) \\ f(12,21,33,43) \\ f(13,21,32,42) \\ f(11,22,32,43) \\ f(12,22,31,42) \\ f(13,22,33,41) \\ f(11,23,33,42) \\ f(12,23,32,41) \\ f(13,23,31,43) \end{pmatrix}$$

Appendix D. Algorithm 2 illustration for $k = 2$ and $n = 9$

creation of the vertices of the unit 2-hypercube and numeration:

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

vertices satisfying the constraints and creation of M_{coor}:

$$\begin{cases} 0 \leq 0 \\ 1 \not\leq 0 \\ 0 \leq 1 \\ 1 \leq 1 \end{cases} \longrightarrow M_{coor} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

vectors of permutations and creation of the 2! matrices by column-wise permutations of M_{coor}:

$$v_1 = \begin{pmatrix} 1 & 2 \end{pmatrix} \longrightarrow M_1 = M_{coor} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 2 & 1 \end{pmatrix} \longrightarrow M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

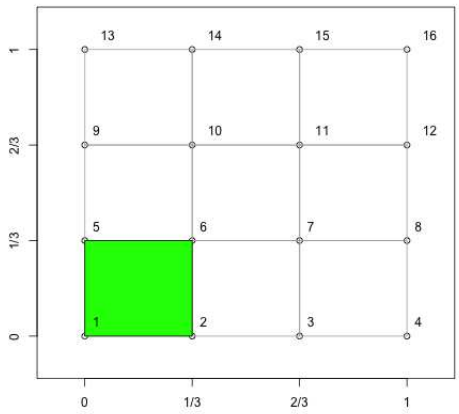
retrieve the vectors of numbers of matching vertices:

$$M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow \begin{matrix} \boxed{1} \\ 2 \\ \boxed{3} \\ \boxed{4} \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow w_1 = \begin{pmatrix} 1 & 3 & 4 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \longrightarrow \begin{matrix} \boxed{1} \\ \boxed{2} \\ 3 \\ \boxed{4} \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow w_2 = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}$$

matrix of subdivision of the 2-hypercube into $3^2 = 9$ sub-hypercubes (one sub-hypercube per row of the matrix):

$$\begin{pmatrix} \boxed{1} & \boxed{2} & 5 & \boxed{6} \\ 2 & 3 & 6 & 7 \\ 3 & 4 & 7 & 8 \\ 5 & 6 & 9 & 10 \\ 6 & 7 & 10 & 11 \\ 7 & 8 & 11 & 12 \\ 9 & 10 & 13 & 14 \\ 10 & 11 & 14 & 15 \\ 11 & 12 & 15 & 16 \end{pmatrix}$$



for each sub-hypercube, for each w_j conserve the vertices whose index corresponds to w_j:

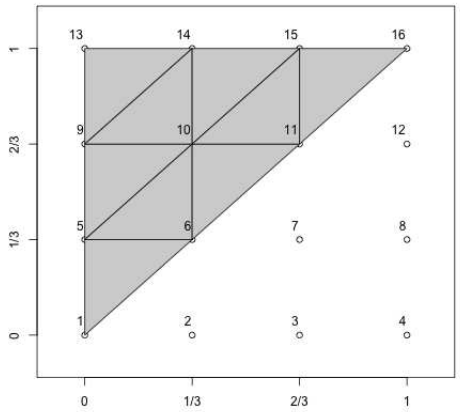
$$\begin{cases} w_1 = \begin{pmatrix} 1 & 3 & 4 \end{pmatrix} \longrightarrow \begin{pmatrix} \boxed{1} & 2 & \boxed{5} & \boxed{6} \end{pmatrix} \\ w_2 = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix} \longrightarrow \begin{pmatrix} \boxed{1} & \boxed{2} & 5 & \boxed{6} \end{pmatrix} \end{cases}$$

iterate the process over the list of sub-hypercubes, to get all the sets of vertices, then conserve only those satisfying the constraints:

$$\begin{cases} w_1 = \begin{pmatrix} 1 & 5 & 6 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \leq 0 \\ 0 \leq \frac{1}{3} \\ \frac{1}{3} \leq \frac{1}{3} \end{pmatrix} \longrightarrow \textit{accept} \\ w_2 = \begin{pmatrix} 1 & 2 & 6 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \leq 0 \\ \frac{1}{3} \not\leq 0 \\ \frac{1}{3} \leq \frac{1}{3} \end{pmatrix} \longrightarrow \textit{reject} \end{cases}$$

this gives a total of $3^2 = 9$ conserved simplices (one simplex per row of the matrix):

$$\begin{pmatrix} 1 & 5 & 6 \\ 5 & 9 & 10 \\ 5 & 6 & 10 \\ 6 & 10 & 11 \\ 9 & 13 & 14 \\ 9 & 10 & 14 \\ 10 & 14 & 15 \\ 10 & 11 & 15 \\ 11 & 15 & 16 \end{pmatrix}$$



We then sample one point in each of the 9 simplices thus created.