Iterative estimation of Sobol' indices based on replicated designs

Laurent Gilquin, Élise Arnaud, Clémentine Prieur, Herve Monod

Recursive estimation procedure of Sobol' indices based on replicated designs

Laurent Gilquin, Elise Arnaud, Clémentine Prieur, Hervé Monod

Recursive estimation procedure of Sobol' indices based on replicated designs

Laurent Gilquin[a,b,*], Elise Arnaud[b], Clémentine Prieur[b], Hervé Monod[c]

[a]*Inria Grenoble - Rhône-Alpes, Inovallée, 655 avenue de l'Europe, 38330 Montbonnot*
[b]*Univ. Grenoble Alpes, Jean Kunzmann Laboratory, F-38000 Grenoble, France CNRS, LJK, F-38000 Grenoble, France, Inria*
[c]*MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-En-Josas, France*

Abstract

In the field of sensitivity analysis, Sobol' indices are widely used to assess the importance of inputs of a model to its output. Among the methods that estimate these indices, the replication procedure is noteworthy for its efficient cost. A practical problem is how many model evaluations must be performed to guarantee a sufficient precision on the Sobol' estimates. This paper tackles this issue by rendering the replication procedure recursive. We consider the ability of adding new points to progressively increase the accuracy of the estimates. The key feature of this approach is the construction of nested space-filling designs. For the estimation of first-order indices, we exploit a nested Latin hypercube already introduced in the literature. For the estimation of closed second-order indices, two constructions of a nested orthogonal array are proposed. Regularity and uniformity properties of the nested designs are studied.

Keywords: sensitivity analysis, Sobol' index, space-filling, orthogonal array, recursive estimator

1. Introduction

Mathematical models used in various fields are often quite complex. The behavior of some of these models may only be explored through the study of uncertainties propagated from their inputs. Sensitivity analysis studies how

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

the uncertainty on an output of a mathematical model can be attributed to sources of uncertainty among the inputs. Among the large number of available approaches, the variance-based method introduced by Sobol' [19] relies on the calculation of sensitivity measures called Sobol' indices. The method is based on a variance decomposition of the model output into fractions which can be attributed to sets of inputs, assuming that the uncertainty on the sets of inputs is modeled by independent probability distributions. The influences of each set are summarized by the Sobol' indices which are scalars between 0 and 1. The higher the index the more influential the set. One can distinguish first-order indices that estimate the main effect of each set of inputs from higher-order indices that estimate the corresponding order of interactions between sets of inputs. Various procedures have been proposed in the literature (see Saltelli [18] for a survey) to estimate Sobol' indices. Unfortunately, these procedures require a significant number of model evaluations that can be prohibitive for expansive models. A solution reducing this number lies in the use of replicated designs.

The notion of replicated designs was introduced by McKay [10]. Later on, Mara *et al.* [9] combine these designs with "pick-freeze" estimators [19] to estimate first-order Sobol' indices. This procedure, called replication procedure, has been further studied and generalized in Tissot *et al.* [21] to the estimation of closed second-order indices. This generalization relies on the construction of orthogonal arrays (OA) (see [5]). The replication procedure has the major advantage of reducing considerably the estimation cost as it requires to construct only two replicated designs of size n. However, if the input space is not properly explored, that is if n is too small, the Sobol' indices estimates may not be accurate enough.

To address this challenge, we need a procedure to sequentially add new points to an initial design and a recursive formula of the Sobol' index estimator. Adding new points is straightforward when the initial design is composed with independent and identically distributed points. However, in the replication procedure as introduced in [21], the initial design possesses either a structure of Latin hypercube or orthogonal array whether first- or closed second-order Sobol' indices are estimated. To preserved these structures, we focus on the construction of nested space-filling designs. An algorithm for the construction of nested Latin hypercubes has been proposed by Qian [14]. It allows to double the size of the design at each step. Our approach to render the replication procedure recursive for the estimation of first-oder Sobol' indices is based on this construction.

The main contributions of this paper are two constructions of a nested orthogonal array for the recursive estimation of closed second-order indices with the replication procedure. Each construction starts with an initial orthogonal array and updates it sequentially by adding a fixed number of new points. Constructions of nested orthogonal arrays have already been studied in [16, 15, 2]. These latter suffer from at least one of the following drawbacks:

- The size of the initial design is rather large, hence at each step a large number of new points is added.
- The constructions deal only with specific values of the input space dimension.
- The discretization is not the same in each dimension, more precisely only one dimension is finely discretized.

Conversely, the two constructions we propose do not suffer from these drawbacks. The paper is organized as follows. In Section 2, backgrounds are given on Sobol' indices and the replication procedure. Then, the process rendering the replication procedure recursive is detailed. Section 3 deals with the construction of the nested space-filling designs: nested Latin hypercube and nested orthogonal arrays. In Section 4, regularity and uniformity properties of these two designs are studied. The end of this section is devoted to an application of our recursive procedure to a toy example.

2. Recursive estimation of Sobol' indices

2.1. Definition of Sobol' indices

Consider the following model defined from a black box perspective:

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

where y is the output of the model f, $\boldsymbol{x}$ the input vector and d the dimension of the input space. Denote by $\subsetneq$ the proper (strict) inclusion symbol and by $\subseteq$ the inclusion symbol.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We model the uncertainty on the inputs by a random vector $\boldsymbol{X} = (X_1, \ldots, X_d)$ whose components are independent. Let $u \subseteq \mathcal{D}$, $\boldsymbol{X}_u$ denotes a vector with components X_j, $j \in u$. Let $P_{\boldsymbol{X}} = P_{X_1} \otimes \ldots \otimes P_{X_d}$ be the distribution of $\boldsymbol{X}$. We assume that $f \in \mathbb{L}^2(P_{\boldsymbol{X}})$.

The model f can then be uniquely decomposed into summands of increasing dimensions (functional ANOVA decomposition [19, 6]):

$$f(\boldsymbol{X}) = f_0 + \sum_j f_j(X_j) + \sum_{k<l} f_{k,l}(X_k, X_l) + \cdots + f_{1,\ldots,d}(X_1, \ldots, X_d) \ , \quad (2)$$

where $\mathrm{E}[f_u(\boldsymbol{X}_u) f_w(\boldsymbol{X}_w)] = 0, \ \forall \ (u, w) \subseteq \{1, \ldots, d\}^2, \ u \neq w$. Denote $Y = f(\boldsymbol{X})$, this implies that $f_0 = \mathrm{E}[Y]$ and that the components are mutually orthogonal with respect to $P_{\boldsymbol{X}}$. Let $u \subseteq \{1, \ldots, d\}$, each component is defined by:

$$f_u(\boldsymbol{X}_u) = \mathrm{E}[Y|\boldsymbol{X}_u] - \sum_{v \subsetneq u} f_v(\boldsymbol{X}_v) \ .$$

The functional decomposition can be used to measure the global sensitivity of the output Y to $\boldsymbol{X}_u$. By squaring and integrating (2), due to orthogonality we get:

$$\sigma^2 = \mathrm{Var}[Y] = \sum_j \sigma_j^2 + \sum_{k<l} \sigma_{k,l}^2 + \cdots + \sigma_{1,\ldots,d}^2 \ , \quad (3)$$

where:

$$\sigma_u^2 = \mathrm{Var}[f_u(\boldsymbol{X}_u)] = \mathrm{Var}[\mathrm{E}[Y|\boldsymbol{X}_u]] - \sum_{v \subsetneq u} \sigma_v^2 \ .$$

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

$$S_u = \frac{\sigma_u^2}{\sigma^2} \ .$$

Let $|u|$ denote the cardinal of u. The Sobol' index S_u measures the contribution to σ^2 of the $|u|^{\text{th}}$-order interaction between the X_j, $j \in u$. Closed Sobol' indices are defined by:

$$\underline{S}_u = \frac{\mathrm{Var}[\mathrm{E}[Y|\boldsymbol{X}_u]]}{\sigma^2} \ .$$

The closed Sobol' index $\underline{S}_u$ measures the contribution of the X_j, $j \in u$, by themselves or in interaction with each other. As an example, if $u = \{k, l\}$, $k \neq l$, then $\underline{S}_{k,l} = S_{k,l} + S_k + S_l$. At last, note that:

$$\sum_{u \subseteq \{1,\ldots,d\}, u \neq \varnothing} S_u = 1 \ ,$$

which gives a direct interpretation of the value of each index. Most of the time, no explicit formulation of Sobol' indices is available. To bypass this problem, one needs to resort to estimation methods.

2.2. Estimation of Sobol' indices

In this section, we review succinctly the estimation procedure introduced by Sobol' [19]. Consider $\boldsymbol{X}$ and $\boldsymbol{X}'$ two independent vectors distributed as the input vector. Let $u \subseteq \mathcal{D}$ and denote by $-u$ its complement. The hybrid point $\boldsymbol{W} = (\boldsymbol{X}_u : \boldsymbol{X}'_{-u})$ is defined by $W_j = X_j$ if $j \in u$ and $W_j = X'_j$ otherwise. We define the following model outputs: $Y = f(\boldsymbol{X})$, $Y_u = f(\boldsymbol{X}_u : \boldsymbol{X}'_{-u})$.

To estimate $\underline{S}_u$, we start from the formula introduced by Janon *et al.* [7, Lemma 1.2] that expresses the Sobol' index as a regression coefficient between the two ouputs Y and Y_u:

$$\underline{S}_u = \frac{\mathrm{Cov}(Y, Y_u)}{\mathrm{Var}[Y]} \ .$$

Then, we proceed as in [19] and introduce two designs, each of size n:

$$\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^n, \qquad \mathcal{P}' = \{\boldsymbol{X}'_i\}_{i=1}^n \ .$$

$\mathcal{P}$ (resp .$\mathcal{P}'$) is a matrix where each row is a point $\boldsymbol{X}_i = (X_{i,1}, \dots, X_{i,d})$ (resp. $\boldsymbol{X}'_i$) of the input space and each column contains n realizations $X_{i,j}$ of each input X_j, $j = 1, \dots, d$. A third design $\mathcal{P}^u = \{\boldsymbol{X}_{i,u} : \boldsymbol{X}'_{i,-u}\}_{i=1}^n$ is constructed from $\mathcal{P}$ and $\mathcal{P}'$ by columns substitution. By evaluating the model with $\mathcal{P}$ and $\mathcal{P}^u$, we obtain n realizations of Y and Y_u noted $\{Y_i\}_{i=1}^n$ and $\{Y_{i,u}\}_{i=1}^n$. Following [11], we consider the estimator:

$$\widehat{\underline{S}}_u = \frac{\frac{1}{n}\sum_{i=1}^n Y_i Y_{i,u} - \left(\frac{1}{n}\sum_{i=1}^n Y_i\right)\left(\frac{1}{n}\sum_{i=1}^n Y_{i,u}\right)}{\frac{1}{n}\sum_{i=1}^n (Y_i)^2 - \left(\frac{1}{n}\sum_{i=1}^n Y_i\right)^2} \ . \tag{4}$$

Other choices are possible for the estimator (see [13] for a succinct review). We focus on (4) whose asymptotic properties have been studied in [7].

The main drawback of the aforementioned procedure is the high number of model evaluations required. Estimating all first-order (resp. all closed second-order) Sobol' indices costs $n(d+1)$ (resp. $n(\binom{d}{2}+1)$) model evaluations. The larger n, the more accurate the estimation of Sobol' indices. Some improvements have been introduced by Saltelli [17] to reduce the number of evaluations but with a cost still depending on the input space dimension. A solution reducing drastically this costs lies in the use of replicated designs.

2.3. Replication procedure and associated designs

In the following, we review the procedure based on replicated designs to estimate first- or closed second-order Sobol' indices. We refer to it as replication procedure. We assume that the inputs $X_1, \ldots, X_d$ are independent and uniformly distributed on $[0,1]$. The generalization to other product distributions is provided in Remark 1 page 8.

The concept of replicated designs was first introduced by McKay in [10]. Here, we define it as follows:

Definition 1. Let $\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\boldsymbol{X}'_i\}_{i=1}^n$ be two designs in $[0,1]^d$. Let $\mathcal{P}^u = \{\boldsymbol{X}_{i,u}\}_{i=1}^n$ (resp. $\mathcal{P}'^u$), $u \subsetneq \mathcal{D}$, denote the subset of dimensions (columns) of $\mathcal{P}$ (resp. $\mathcal{P}'$) indexed by u. We say that $\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order $a \in \{1, \ldots, d-1\}$ if for any $u \subsetneq \mathcal{D}$ such that $|u| = a$, $\mathcal{P}^u$ and $\mathcal{P}'^u$ are the same point set in $[0,1]^a$. We define by π_u the permutation that rearranges the rows of $\mathcal{P}'^u$ into $\mathcal{P}^u$.

Example. Consider the two designs:

$$\mathcal{P} = \begin{pmatrix} 0.08 & 0.46 & 0.21 \\ 0.15 & 0.77 & 0.43 \\ 0.89 & 0.30 & 0.05 \\ 0.70 & 0.23 & 0.95 \end{pmatrix}, \qquad \mathcal{P}' = \begin{pmatrix} 0.89 & 0.30 & 0.95 \\ 0.15 & 0.23 & 0.21 \\ 0.70 & 0.46 & 0.43 \\ 0.08 & 0.77 & 0.05 \end{pmatrix}.$$

$\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order 1. $\forall\, i$, the i-th columns of $\mathcal{P}$ and $\mathcal{P}'$ share the same unordered set of values. The permutation $\pi_1 = (4, 2, 1, 3)$ order the first column of $\mathcal{P}'$ into the first column of $\mathcal{P}$.

The key point of the replication procedure is to construct two replicated designs and use their structure to mimic the hybrid points $\{\boldsymbol{X}_{i,u} : \boldsymbol{X}'_{i,-u}\}_{i=1}^n$. More precisely, let $\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\boldsymbol{X}'_i\}_{i=1}^n$ be two replicated designs of order $|u|$. Denote by $\{Y_i\}_{i=1}^n$ and $\{Y'_i\}_{i=1}^n$ the two sets of model evaluations obtained with $\mathcal{P}$ and $\mathcal{P}'$. From Definition 1, we know that $\boldsymbol{X}'_{\pi_u(i),u} = \boldsymbol{X}_{i,u}$.

$$\begin{aligned} \text{Then, } Y'_{\pi_u(i)} &= f(\boldsymbol{X}'_{\pi_u(i),u} : \boldsymbol{X}'_{\pi_u(i),-u}) \ , \\ &= f(\boldsymbol{X}_{i,u} : \boldsymbol{X}'_{\pi_u(i),-u}) \ . \end{aligned}$$

Hence, each Sobol' index $\underline{S}_u$ can be estimated via formula (4) with $Y'_{\pi_u(i)}$ in place of $Y_{i,u}$ without requiring further model evaluations. As such, the cost of the procedure equals $2n$ to estimate either all first-order or all closed second-order indices. We detail below the structure of the two replicated designs for each case.

Estimation of first-order indices. There are various choices for constructing two replicated designs of order 1. In [9], $\mathcal{P}$ and $\mathcal{P}'$ are composed with i.i.d points. In [21], the authors propose to use Latin hypercube designs insuring most of the time a better exploration of the input space:

Definition 2 (Latin hypercube design). Denote by Π_n the set of all the permutations of $\{1,\dots,n\}$ and let $\pi_1,\dots,\pi_d$ be d independent random variables uniformly distributed on Π_n. $\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^n$ is a Latin hypercube design if:

$$\boldsymbol{X}_i = \left(\frac{\pi_1(i) - U_{i,1}}{n}, \dots, \frac{\pi_d(i) - U_{i,d}}{n}\right), \tag{5}$$

where the $U_{i,j}$ are independent random variables uniformly distributed on $[0,1]$ and independent of the π_j.

The first replicated design $\mathcal{P}$ is constructed with Definition 2 above, then $\mathcal{P}'$ is obtained by permuting independently the values of each column of $\mathcal{P}$. In [21], the two resulting designs are referred as replicated Latin hypercube designs.

Estimation of closed second-order indices. The generalization to second-order indices was introduced in [21]. To estimate these indices, one needs to find a structure that "freezes" each subset of two variables. A solution relies on the use of orthogonal arrays [5, Definition 1.1]:

Definition 3 (Orthogonal array). A $n \times d$ array $A = \{\boldsymbol{A}_i\}_{i=1}^n$, $\boldsymbol{A}_i = (A_{i,1},\dots,A_{i,d})$, with values from a set S of cardinality q is said to be an orthogonal array with q levels, strength t $(0 \le t \le d)$ and index λ if every $n \times t$ sub-array of A contains each t-tuple based on S exactly λ times as a row. The orthogonal array A satisfies $n = \lambda q^t$. It is denoted by $OA_\lambda(q,d,t)$.

Here, the space S is identified as the Galois field of order q, denoted by $GF(q)$, where q is a prime number or prime power number ($q = p^\alpha$, p prime and $\alpha \in \mathbb{N}$). Once an orthogonal array is constructed its levels are substituted by $1,\dots,q$, q indicating the number of points into which each input is discretized. For constructions of orthogonal arrays we invite the reader to consult [5].

From Definition 3, orthogonal arrays of strength two are replicated designs of order 2 and as such naturally "freeze" each subset of two variables.

Therefore, the strategy proposed in [21] is to construct two replicated orthogonal arrays of strength two:

Definition 4 (Replicated orthogonal arrays). Let $A = \{\boldsymbol{A}_i\}_{i=1}^{q^t}$ be an $OA_1(q, d, t)$. Denote by Π_q the set of all the permutations of $\{1, \ldots, q\}$ and let $\pi_1, \ldots, \pi_d$ be d independent random variables uniformly distributed on Π_q. $\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^{q^t}$ and $\mathcal{P}' = \{\boldsymbol{X}'_i\}_{i=1}^{q^t}$ are two replicated orthogonal arrays if:

$$\begin{aligned} \boldsymbol{X}_i &= \left(\frac{A_{i,1} - U_{i,1}}{q}, \ldots, \frac{A_{i,d} - U_{i,d}}{q} \right), \\ \boldsymbol{X}'_i &= \left(\frac{\pi_1(A_{i,1}) - U_{\pi_1(A_{i,1}),1}}{q}, \ldots, \frac{\pi_d(A_{i,d}) - U_{\pi_d(A_{i,d}),d}}{q} \right), \end{aligned} \tag{6}$$

where the $U_{i,j}$ are independent random variables uniformly distributed on $[0, 1]$ and independent of the π_j.

Note that this definition is given in a Monte-Carlo context (the points lies in $[0, 1]^d$). The construction of $\mathcal{P}'$ reduces to apply independently a permutation to the symbols of each column of A. We note by $\Diamond$ the operator achieving this rearrangement:

$$A' = \Diamond(A, \{\pi_1, \ldots, \pi_d\}) \Leftrightarrow \boldsymbol{A}'_i = (\pi_1(A_{i,1}), \ldots, \pi_d(A_{i,d})), \; i = 1, \ldots, n. \tag{7}$$

Using two replicated orthogonal arrays of strength two, the cost of the replication procedure writes $2n$ where $n = q^2$.

Remark 1. In this section, the constructions of designs $\mathcal{P}$ and $\mathcal{P}'$ are only valid when dealing with variables $X_1, \ldots, X_d$ independent and uniformly distributed on $[0, 1]$. However, these constructions can be generalized to other non-uniform distributions. Denote by $F_1, \ldots, F_d$ the cumulative distribution functions of $X_1, \ldots, X_d$. In the general case, the two designs $\mathcal{P} = \{\boldsymbol{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\boldsymbol{X}_i\}_{i=1}^n$ are constructed as follows:

$$\boldsymbol{X}_i = \left(F_1^{-1}(X_{i,1}), \ldots, F_d^{-1}(X_{i,d})\right),$$

$$\boldsymbol{X}'_i = \left(F_1^{-1}(X'_{i,1}), \ldots, F_d^{-1}(X'_{i,d})\right),$$

where $F_1^{-1}, \ldots, F_d^{-1}$ are the quantile functions of $X_1, \ldots, X_d$.

2.4. Recursive procedure

To estimate Sobol' indices with one of the latter estimation methods, one needs to fix a value for the size n of the designs. In practice, one wants to choose n large enough to ensure a sufficient precision on the Sobol' estimates while keeping an affordable computational time. This choice is difficult to address mostly because it depends on the complexity of the model studied. As such, it is hard to bring out a general rule of thumb.

The practical solution investigated here is to increase the accuracy of the estimates by sequentially adding new points. To do so, we propose a recursive version of the replication procedure. First, the construction of the two replicated designs is carried out according to the following scheme:

$$\left\{ \begin{array}{l} \mathcal{P}_0 = B_0 \\ \mathcal{P}_\ell = \mathcal{P}_{\ell-1} \cup B_\ell \end{array} \right. , \quad \left\{ \begin{array}{l} \mathcal{P}'_0 = B'_0 \\ \mathcal{P}'_\ell = \mathcal{P}'_{\ell-1} \cup B'_\ell \end{array} \right. , \quad \ell \geq 1 ,$$

where B_ℓ, B'_ℓ are the new sets of m_ℓ points added at step ℓ. In the following, we refer to these sets as blocks. The blocks B_ℓ and B'_ℓ are then used to evaluate the model and obtain two sets of outputs $\{Y^i\}_{i=n_{\ell-1}+1}^{n_\ell}$ and $\{Y_u^i\}_{i=n_{\ell-1}+1}^{n_\ell}$ where n_ℓ denote the size of designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$. Therefore, these two designs are nested designs partitioned into blocks.

Our recursive procedure requires to write down a recursive formula for the Sobol' index estimator. Recall the expression of the Sobol' index:

$$\underline{S}_u = \frac{\mathrm{Cov}[Y, Y_u]}{\mathrm{Var}[Y]} = \frac{\mathrm{E}[Y Y_u] - \mathrm{E}[Y]\ \mathrm{E}[Y_u]}{\mathrm{Var}[Y]} , \tag{8}$$

where Y and Y_u are evaluated with the two replicated designs. At step $\ell \geq 1$, the Sobol' index $\underline{S}_u$ is estimated by the following family of recursive estimators:

$$\widehat{\underline{S}}_u^{(\ell)} = \frac{\phi_\ell - \psi_\ell \xi_\ell}{V_\ell} , \tag{9}$$

where ϕ_ℓ, ψ_ℓ, ξ_ℓ and V_ℓ are identified with formula (8) and are estimated at step $\ell = 0$ directly from blocks B_0 and B'_0. These terms are recursively defined as follows:

$$
\begin{cases}
n_\ell = n_{\ell-1} + m_\ell \ , \\
n_\ell \phi_\ell = n_{\ell-1}\phi_{\ell-1} + m_\ell \displaystyle\sum_{i=n_{\ell-1}+1}^{n_\ell} Y^i Y_u^i \ , \\
n_\ell \psi_\ell = n_{\ell-1}\psi_{\ell-1} + m_\ell \displaystyle\sum_{i=n_{\ell-1}+1}^{n_\ell} Y^i \ , \\
n_\ell \xi_\ell = n_{\ell-1}\xi_{\ell-1} + m_\ell \displaystyle\sum_{i=n_{\ell-1}+1}^{n_\ell} Y_u^i \ , \\
(n_\ell - 1)V_\ell = (n_\ell - 2)V_{\ell-1} + n_{\ell-1}\psi_{\ell-1}^2 + \displaystyle\sum_{i=n_{\ell-1}+1}^{n_\ell} (Y^i)^2 - n_\ell \psi_\ell^2 \ .
\end{cases}
$$

Algorithm 1 summarizes the main steps of our recursive procedure. The form of the stopping criterion (variable `test`) is discussed in Section 4.2. The set $\mathcal{D}$ equals either $\{1, \ldots, d\}$ or $\{(k, l) \in \{1, \ldots, d\}^2;\ k < l\}$ whether first-order or closed second-order indices are estimated. The cost of our recursive procedure equals $2 \times \sum_{\ell \geq 0} m_\ell$.

Algorithm 1 Recursive estimation of Sobol' indices

$\ell \leftarrow 0$, $\widehat{\underline{S}}_u^{(0)} \leftarrow 0$, `test` $\leftarrow$ true
$\mathcal{P}_0 \leftarrow B_0$
$\mathcal{P}'_0 \leftarrow B'_0$
while `test` **do**
  for $u \subset \mathcal{D}$ **do**
    Compute Y and Y_u from B_l and B'_l
    Evaluate $\widehat{\underline{S}}_u^{(\ell)}$ with (9)
  end for
  `test` $\leftarrow$ stopping criterion
  $\mathcal{P}_{\ell+1} \leftarrow \mathcal{P}_\ell \cup B_{\ell+1}$
  $\mathcal{P}'_{\ell+1} \leftarrow \mathcal{P}'_\ell \cup B'_{\ell+1}$
  $\ell \leftarrow \ell + 1$
end while
Return the Sobol' estimates

In the next section, we detail the construction of the two nested designs required for the estimation of either first-order or closed second-order Sobol' indices. In both cases, the construction ensures that at each step $\ell \geq 0$,

$\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ possess a space-filling structure and are two replicated designs or order 1 or 2.

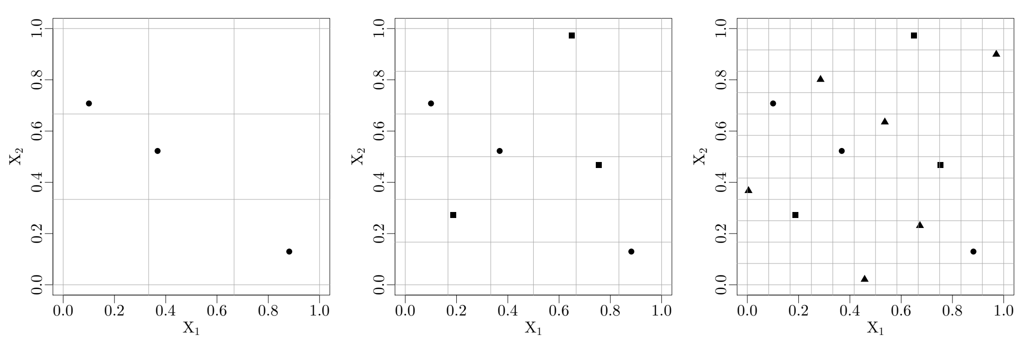
3. Nested space-filling designs

For the estimation of first-oder indices, the two nested designs are nested Latin hypercube designs. The number of blocks partitioning the design has to be specified beforehand. This number defines the initial discretization of the input space. This discretization is then further refined with the addition of a new block. For the estimation of closed second-order indices, the two nested designs are nested orthogonal arrays of strength two. The number of blocks is iteratively augmented. However, the discretization of each input is fixed at the first step and remains unvaried throughout the procedure.

3.1. Nested Latin hypercube design

A way to augment the number of points while preserving a Latin hypercube structure has been proposed by Qian in [14]. A nested Latin hypercube design is a Latin hypercube design partitioned into blocks defining multiple layers. As an illustration, a two dimensional nested Latin hypercube design with 3 layers is presented in Figure 1. Each layer possesses itself a Latin hypercube structure in a grid progressively refined. The algorithm underlying the construction of a nested Latin hypercube design is detailed in [14, Section 5].

Figure 1: Nested Latin hypercube design with three layers (a), (b), (c). The symbols mark the new points (i.e the blocks) added at each step (in order circle, square, triangle).



(a) first layer (b) second layer (c) third layer

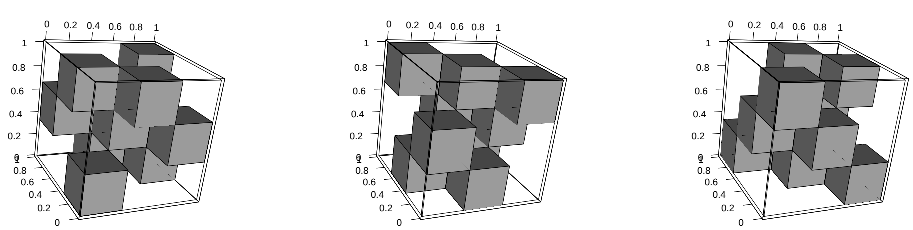
For the estimation of first-order indices, our two nested designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are two nested Latin hypercube designs. First, the block B_ℓ is constructed using the algorithm in [14]. Then, B'_ℓ is obtained by permuting independently the values in each column of B_ℓ. This guarantees that at each step $\ell \geq 0$, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ possess a structure of Latin hypercube and are two replicated designs of order 1. Using the construction proposed in [14], the size of $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ equals at least 2^ℓ.

3.2. Nested orthogonal array of strength two

The nested designs constructed for the estimation of closed second-order indices are two nested orthogonal arrays of strength two. Here, a nested orthogonal array of strength two corresponds to an $OA_\lambda(q, d, 2)$ where $\lambda > 1$. It can be partitioned into λ blocks where each block has the geometric structure of an $OA_1(q, d, 2)$. We propose two methods to construct such designs. The first one is an *accept-reject* method. The second one is called *algebraic* method and relies on results from arithmetic.

The idea of each construction is to progressively fill the d-hypercube (the discretized input space) with distinct sub-hypercubes where each sub-hypercube corresponds to a row of a new block. As an illustration, consider the example of an $OA_3(3, 3, 2)$ represented in Figure 2 below. This OA is

Figure 2: $OA_3(3, 3, 2)$ with 3 blocks. (a) first block. (b) second block. (c) third block. Each block can be identified as an $OA_1(3, 3, 2)$. Points are represented by their respective sub-hypercube.



(a) first block (b) second block (c) third block

partitioned in three blocks represented in graphs (a), (b), (c). Each block possesses the structure of an $OA_1(3, 3, 2)$. These three blocks form a partition of the hypercube. The *accept-reject* and the *algebraic* methods strive to construct both an $OA_\lambda(q, d, 2)$ where the rows are two by two distinct. Two rows are said distinct if they differ in at least one component. The idea is to evaluate the model on previously unexplored regions of the input space.

Each method starts with the construction of an initial $OA_1(q, d, 2)$ noted A_0. Then at each step $\ell \geq 1$, a new $OA_1(q, d, 2)$, noted A_ℓ, is constructed. The two blocks B_ℓ and B'_ℓ are obtained from A_ℓ as in Definition 4. As a result, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ both have a structure of $OA_\ell(q, d, 2)$ and are replicated designs of order 2. The process is repeated until the stopping criterion is met. The form of the stopping criterion is discussed in Section 4.2.

The *accept-reject* and the *algebraic* methods differ on the way A_ℓ is constructed. We detail below this step for each method.

Method 1: accept-reject. Variant 1 details the construction of A_ℓ. It uses the operator $\Diamond$ defined in Definition 7. The idea is to randomly construct a new orthogonal array from A_0 using $\Diamond$ and test if its rows are distinct from those of each previous orthogonal array constructed; namely $A_{\ell-1}, A_{\ell-2}, \ldots, A_0$. This test may become computationally expensive for small input space dimension as the probability of acceptation decreases faster.

Variant 1 *Accept-reject* method for the construction of A_ℓ

Set `bool` $\leftarrow$ false
while !`bool` **do**
    Sample $\pi_1, \ldots, \pi_d$ in Π_q
    Construct $A_\ell = \Diamond(A_0, \{\pi_1, \ldots, \pi_d\})$ with (7)
    for $k = 0, \ldots, \ell - 1$ **do**
        $\texttt{bool}_k \leftarrow rows(A_\ell) \cap rows(A_k) == \varnothing$
    end for
    `bool` $\leftarrow \forall k : \texttt{bool}_k$
end while

Method 2: Algebraic method. Define the following set:

$$C = \Big\{ g = (0, 0, g_3, \ldots, g_d) \mid \forall i \geq 3,\ g_i \in GF(q) \Big\} \ \subsetneq\ GF(q)^d \ .$$

Variant 2 details the construction of A_ℓ. $\oplus$ is the addition in $GF(q)^d$. The idea of the method is to construct a partition of the discretized input space and select A_ℓ from this partition. A_ℓ is viewed as a coset of A_0 and

Variant 2 *Algebraic* method for the construction of A_ℓ

1: Choose $g_\ell \in C$
2: Construct $A_\ell = g_\ell A_0 = \{g_\ell \oplus A_{0i}\}_{i=1}^{q^2}$, $A_{0i} = (A_{0i,1}, \ldots, A_{0i,d})$
3: $C \leftarrow C \setminus \{g_\ell\}$

is obtained using the set C. The main advantage of this method is that the maximum value taken by ℓ is known beforehand (consequence of Proposition 1 thereafter). The following proposition guarantees that A_ℓ constructed in Variant 2 is an $OA_1(q, d, 2)$:

Proposition 1. Consider A_0 an $OA_1(q, d, 2)$ based on $GF(q)^d$. We have the following results:

i) $\forall g \in GF(q)^d$, gA_0 is an $OA_1(q, d, 2)$
ii) $\forall g, g' \in C$, such that $g \neq g'$, $gA_0 \cap g'A_0 = \varnothing$. In other words, the sets $\{gA_0\}$ form a partition of $GF(q)^d$.

Proof. i) Let $g = (g_1, \ldots, g_d) \in GF(q)^d$. Consider A_{0k}, A_{0l} two columns of A_0. Denote by E the group $(GF(q), +)$. Since $g_k E \times g_l E$ is isomorph to $E \times E$, the 2-tuples $(A_{0i,k} + g_k, A_{0i,l} + g_l)$ obtained after addition are all two by two distinct.

ii) The proof can be found in [20] where an orthogonal array is regarded as a "systematic linear code".

□

As a consequence of ii), the maximum number of blocks one can construct using the *algebraic* method equals the cardinality of C, that is q^{d-2}. If this maximum value is reached, the blocks $A_0, A_1, \ldots, A_{q^{d-2}-1}$ form a partition of the discretized input space.

The cost of our recursive procedure to estimate all closed second-order indices equals $2 \times K \times q^2$ where K is the ending step.

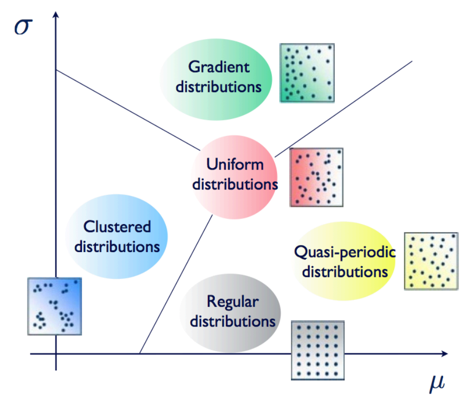
4. Space-filling properties and application

We propose first to study the space-filling properties of the nested design used in our recursive procedure. Then, an application of our recursive procedure is presented on a toy example.

4.1. Space-filling properties

Three criteria are selected to study the properties of the nested designs: the maximin [8], the emst (euclidean minimal spanning tree [3]) and the L^2 star discrepancy [12]. The maximin criterion returns the minimum of the distances between all pairs of points of a design. It can be interpreted as follows: the higher the value, the more regular the scattering of design points. The emst criterion can be interpreted using a (μ, σ) graph, Figure 3, called interpretation graph. A minimal spanning tree is constructed from the

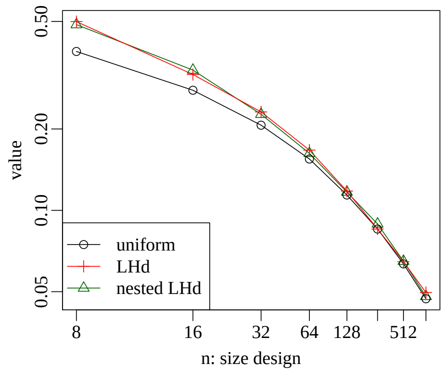
Figure 3: Interpretation graph of the emst. The uniform distribution is used as a reference.



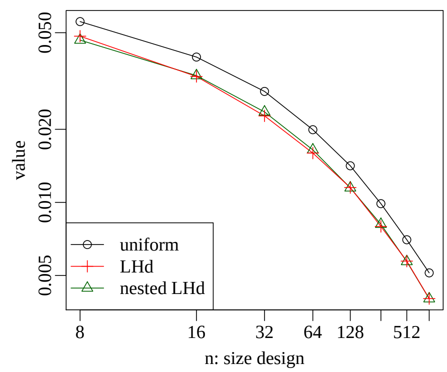
design, then mean (μ) and standard deviation (σ) of the tree edges lengths are evaluated. A value of the emst criterion is represented as a point in the (μ, σ) graph. The uniform distribution, that is i.i.d sampling, is used as a reference. A design having a higher value for μ and a smaller value for σ than those of a uniform design is more regular. Maximin and emst criteria provide together a good estimation of the regularity properties of a design. The L^2 star discrepancy criterion measures the uniformity property of a design. The smaller the value, the more uniform is the design.

We first study properties of the nested Latin hypercube design (nested LHd) used for the estimation of first-order Sobol' indices. Its properties are compared to those of the following designs: (i) uniform design (obtained through i.i.d sampling) and (ii) Latin hypercube design (LHd). Figure 4 shows the results obtained with each of the three criteria. The results are averaged over $r = 100$ repetitions. The input space dimension d equals 5. The

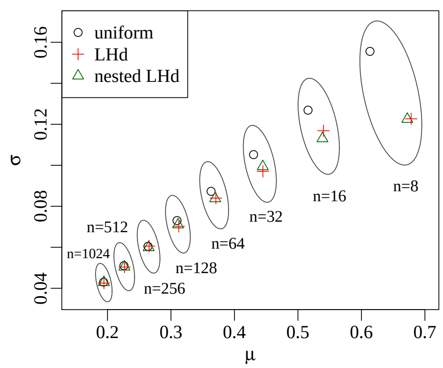
Figure 4: Averaged results of maximin, emst and star discrepancy criteria over 100 repetitions for different sizes n of the designs used for the estimation of first-order indices.



(a) Log-log graph of maximin



(b) Log-log graph of L^2 star discrepancy



(c) emst

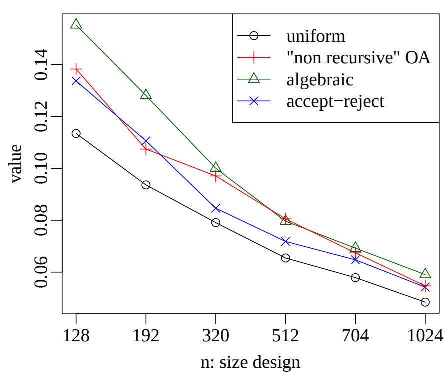
comparison is made for the following sizes n of each design: $(2^3, 2^4, \ldots, 2^{10})$. For the nested LHd, these sizes correspond to those of design $\mathcal{P}_\ell$ augmented over 8 consecutive steps. Both the LHd and the nested LHd give similar results for the three criteria. Furthermore, both designs give better results than the uniform design. As such, in terms of space-filling properties of the designs, there is no drawback to render the replication procedure recursive.

Remark 2. One other class of designs well suited for the estimation of first-order Sobol' indices are low discrepancy sequences. These sequences are points sets sampled so as to approximate as close as possible a uniform distribution and are known to achieve both uniformity and regularity properties. Such sequences could be used in our recursive procedure in place of nested

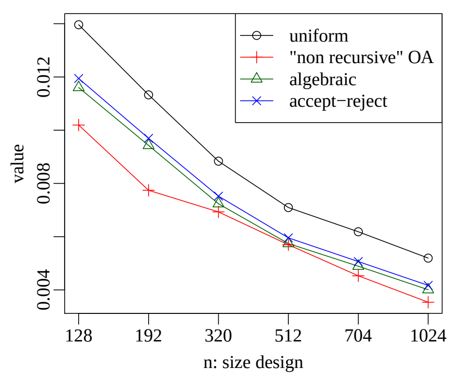
Latin hypercube designs. This alternative has recently been studied in [4].

A second comparison is carried out between the following designs used for the estimation of closed-second order indices: (i) uniform design, (ii) “non-recursive” OA, (iii) *accept-reject* and (iv) *algebraic*. Design (ii) refer to the orthogonal array used in [21]. Designs (iii) and (iv) refers to the design $\mathcal{P}_\ell$ constructed with either the *accept-reject* or the *algebraic* method. Results are again averaged over $r = 100$ repetitions and the input space dimension still equals 5. Figure 5 shows the results obtained with each of the three criteria.

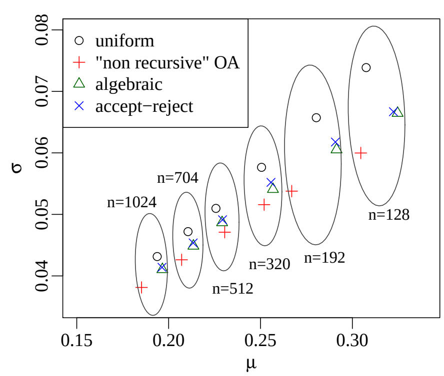
Figure 5: Averaged results of maximin, emst and star discrepancy criteria over 100 repetitions for different sizes n of the designs used for the estimation of closed second-order indices.



(a) Maximin



(b) L^2 star discrepancy



(c) emst

For the sake of visualization, results for only the following sizes n of the designs are represented: $(3 \times 8^2, 5 \times 8^2, 8 \times 8^2, 11 \times 8^2, 15 \times 8^2, 18 \times 8^2)$. In terms of emst and discrepancy criteria, the "non-recursive" OA gives the best results while results for the *accept-reject* and *algebraic* designs are similar. The *algebraic* design gives better results for the maximin criterion than the *accept-reject* design.

The main conclusion is that the *algebraic* design possesses regularity and uniformity properties overall slightly better than those of the *accept-reject* design. These two designs possess slightly worse space-filling properties than their counterpart used in [21]. This difference can be explained by the lack of progressive discretization of the inputs in both the *algebraic* and the *accept-reject* method. However, that is largely offset by the possibility to perform a recursive estimation of the indices.

4.2. Application to a toy example

Our recursive procedure is tested and compared to the classic replication procedure with the Bratley *et al.* function [1], defined as follows:

$$f(X_1, \ldots, X_d) = \sum_{i=1}^{d} (-1)^i \prod_{k=1}^{i} X_k \ ,$$

where $X_1, \ldots, X_d$ are independent random variables uniformly distributed on $[0, 1]$. Both first- and closed second-order Sobol' indices of the function are estimated with each procedure. Both procedure are repeated $r = 100$ times to get samples of estimates. We choose $d = 6$ for the input space dimension. Since f has an analytical expression, theoretical values of the Sobol' indices can be precisely calculated through symbolic integrals evaluations.

Stopping criterion. Our recursive procedure is carried out until a stopping criterion is reached. At each step ℓ of the procedure, the following quantity is evaluated:

$$\mathbf{e}^{(\ell)} = \left|\left| \widehat{\underline{S}}_u^{(\ell)} - \widehat{\underline{S}}_u^{(\ell-1)} \right|\right| ,$$

where $||.||$ denotes the absolute value function. $\mathbf{e}^{(\ell)}$ is an absolute difference between two successive estimations of $\underline{S}_u$. The stopping criterion we proposed is composed of two conditions c_1 and c_2. The first condition c_1 reads as follows:

$$\forall \ u \in \mathcal{D} : \mathbf{e}^{(\ell-\ell_0)} < \varepsilon, \ \mathbf{e}^{(\ell-\ell_0-1)} < \varepsilon, \ldots, \mathbf{e}^{(\ell)} < \varepsilon$$

where $\mathcal{D}$ equals either $\{1, \ldots, d\}$ or $\{(k,l) \in \{1, \ldots, d\}^2;\ k < l\}$ depending on whether first-order or closed second-order Sobol' indices are estimated and $\ell_0 > 0$ is an integer. Condition c_1 tests if all quantities $\mathbf{e}^{(\ell)}$ are smaller than a tolerance ε on ℓ_0 consecutive steps. The second condition c_2 tests if $\ell > \ell_{max}$, where ℓ_{max} is a maximum number of iterations. The parameters ε, ℓ_0 and ℓ_{max} have to be properly set.

The recursive procedure stops when one of the two conditions is verified. From the r repetitions, a vector $(r_1, \ldots, r_K, \ldots, r_{\ell_{max}})$ is constructed where r_K denotes the number of time our recursive procedure has stopped at step K and $r_{\ell_{max}}$ denotes the number of times condition c_2 has been reached. We note by r_α the median of this vector and α the corresponding step.

To have a fair comparison, $\underline{S}_u$ is also estimated r times with the classic replication procedure where the size of the two replicated designs equals the size of $\mathcal{P}_\alpha$ and $\mathcal{P}'_\alpha$ in our recursive procedure.

Estimation of first-order indices. We consider the context where a limited number of evaluation points is available as it is often the case in industrial applications. Therefore, a small value for ℓ_{max} is selected to highlight that our recursive procedure can perform as well as the classic one for a restricted budget of evaluation points. The parameters of the stopping criterion for the recursive procedure are set as follows: $\varepsilon = 0.15$, $\ell_0 = 2$ and $\ell_{max} = 9$. The size of designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ is instantiated at 2^2 and can raise up to $2^{\ell_{max}}$.

Figure 6 shows a barplot representation of the r_K obtained. We observe that condition c_2 is only reached one third of the time. Figure 7 shows the boxplots representation of the estimates obtained with the two replication procedures: recursive (right boxplots) and classic (left boxplots).

The two methods give overall similar results. Hence, there is no drawback to render the replication procedure recursive for the estimation of first-order indices. Furthermore, our recursive procedure shows that the number of model evaluations can be decreased by adopting a sequential approach. One can calculate the gain in terms of number of evaluations. This gain corresponds to the ratio of the maximum number of evaluations $r_{\ell_{max}}$ divided by the iteration at which our recursive procedure stopped. For this example the median gain equals $9/8 = 1.125$ and the maximum gain equals $9/7 = 1.29$.

Estimation of closed second-order indices. The parameters of the stopping criterion are set as follows: $\varepsilon = 3 \times 10^{-3}$, $\ell_0 = 3$ and $\ell_{max} = 100$. The initial

Figure 6: Distribution of the r_K for the estimation of first-order indices. The bar associated to the step α is colored in black.

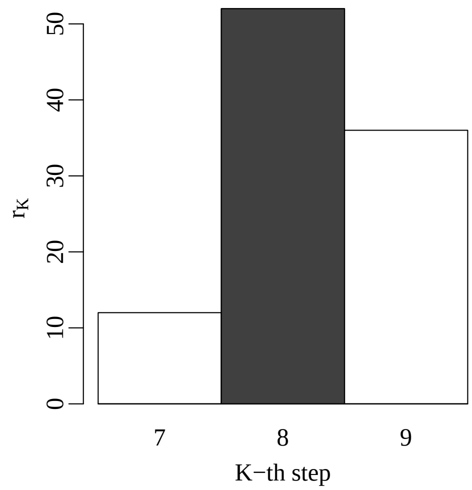
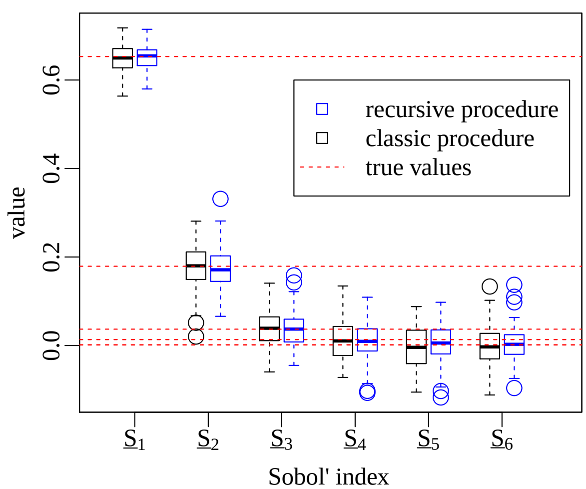


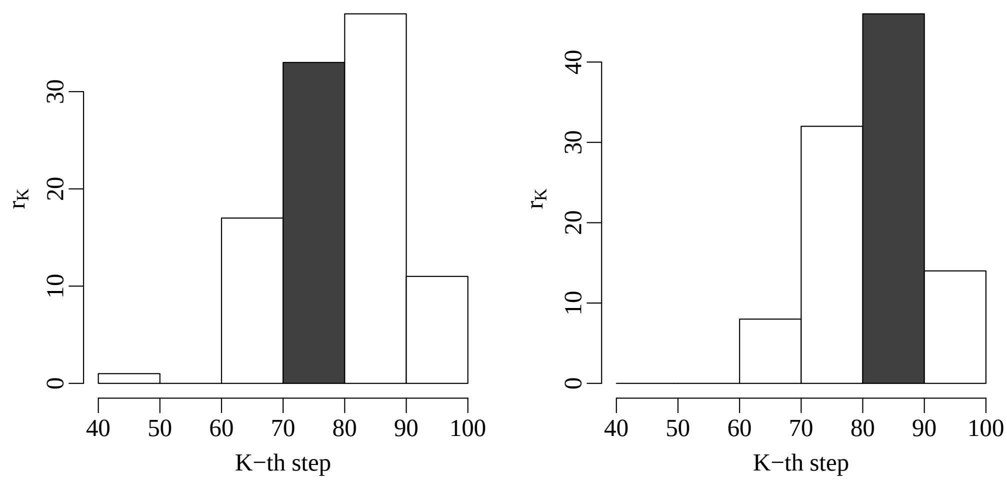
Figure 7: Boxplots of first-order Sobol' indices estimated $r = 100$ times with both our recursive procedure (left boxplot) and the classic replication procedure (right boxplot). The dotted horizontal lines refer to the true values of the indices. True values of indices $\underline{S}_5$ and $\underline{S}_6$ are identical.



orthogonal array A_0 used to augment designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ is constructed by setting $q = 8$. The sizes of these designs can range from 8^2 up to 100×8^2.

Figure 8 shows barplots representation of the r_K obtained when applying our recursive procedure with either the *algebraic* method or the *accept-reject* method. Results show that our recursive procedure finishes at earlier steps when using the *algebraic* method.

Figure 8: Distribution of the r_K when our recursive procedure is applied with either (a) the *algebraic* method or (b) the *accept-reject* method. For each graph, the bar associated to the median step α is colored in black.



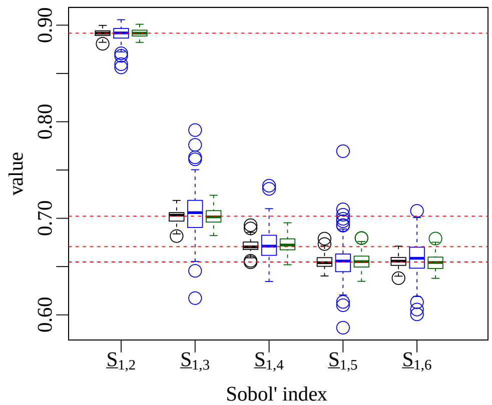
(a) *algebraic* method (b) *accept-reject* method

Figure 9 gives the boxplots representation of the estimates obtained with the classic replication procedure (left boxplots) and with our recursive procedure using either the *algebraic* method (middle boxplots) or the *accept-reject* method (right boxplots).

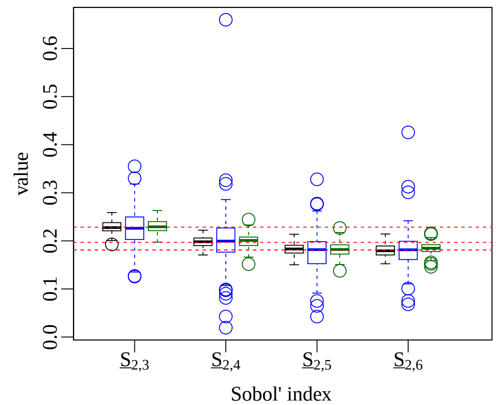
The main observation is that our recursive procedure using the *algebraic* method shows more variability in the estimates than the two others. This observation is emphasized for graphs (b) and (c) of Figure 9 corresponding to Sobol' indices with low values. However, this variability is mostly due to the *algebraic* method itself stopping at earlier steps than the *accept-reject* method. The results obtained with our recursive procedure using the *accept-reject* method are overall similar to those obtained with the classic replication procedure.

As for the case of first-order indices, one can calculate the gain of our recursive procedure in terms of number of evaluations. Table 1 gives the gain

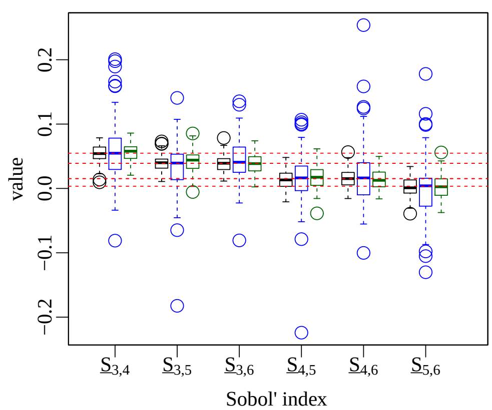
Figure 9: Boxplots of closed second-order Sobol' indices estimated $r = 100$ times with our recursive procedure and the classic replication procedure. For each index $\underline{S}_u$, the left boxplot refers to the classic replication procedure, the boxplot in the middle (resp. on the right) refers to our recursive procedure using the *algebraic* (resp. *accept-reject*) method. The horizontal dotted lines refer to the true values of the indices.



(a) high value indices



(b) low value indices



(c) near zero value indices

of our method for each quartile of the vector $(r_1, \ldots, r_{\ell_{max}})$. Our recursive procedure shows that it is possible to decrease even more the number of simulations by adopting a sequential approach for the estimation of closed second-order indices while preserving the same order of precision. However, as stated before, there is a computational price to pay induced by the *accept-reject* method. When the input space dimension is small ($d \leq 4$), it is harder

Table 1: Gain of the recursive replication procedure using either the *algebraic* or the *accept-reject* construction. The gain is calculated in terms of number of evaluations for each quartile $(r_{1/4}, r_{1/2}, r_{3/4})$ of the vector $(r_1, \ldots, r_{\ell_{max}})$.

quartile	construction	value	gain= $\frac{r_{\ell_{max}}}{\text{value}}$
$r_{1/4}$	*algebraic*	73	1.37
	accept-reject	76	1.32
$r_{1/2}$	*algebraic*	80	1.25
	accept-reject	82	1.22
$r_{3/4}$	*algebraic*	87	1.15
	accept-reject	88	1.14

to find new blocks. As such, the *algebraic* method should be preferred to the *accept-reject* one. At the opposite, when the input space dimension is high, new blocks are easier to find. Therefore, the *accept-reject* method should be used as it gives more accurate results.

Conclusion

In this paper we proposed a new approach rendering the replication procedure recursive to estimate first-order or closed second-order Sobol' indices. We introduced a recursive formula for the Sobol' index estimator. The recursive procedure presented consists in augmenting the two replicated designs with new sets of points through the construction of nested space-filling designs. For the case of closed second-order indices, two methods were proposed to construct a nested orthogonal array of strength two: an *algebraic* method and an *accept-reject* method. Our recursive procedure was compared to the classic replication procedure of Tissot and Prieur [21]. The comparison focused on the space-filling properties of the designs and on the precision of the Sobol' indices estimates.

The replication procedure proposed in [21] are known to be highly efficient in terms of number of simulations. Yet the results in this paper showed that it is still possible to decrease the number of simulations by adopting a sequential procedure based on a recursive method of estimation. More precisely, the nested designs proposed here gave the same order of precision on Sobol' indices as the replicated designs used in [21] but with a random number of simulations of much smaller expectation. Furthermore, the space-filling properties of the nested designs constructed were on average as good as the one of the replicated designs used in [21].

For the case of first-order indices, considering Sobol' sequences could improve the nested designs [4]. For the case of closed second-order indices, the variability in the results showed by our recursive procedure while using the *algebraic* method could be reduced by further working on the set C (Section 3.2 Variant 2). In our case, the set C was filled with elements g chosen at random. A more deterministic choice of the g could lead to a better exploration of the input space.

Acknowledgments

This work is supported by the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020).

References

[1] Bratley, P., Niederreiter, B. L., 1992. Implementation and tests of low-discrepancy sequences. ACM Trans. Model. Comput. Simul. 2 (3), 195–213.

[2] Dey, A., 2012. On the construction of nested orthogonal arrays. Australas. J. Combin. 54, 37–48.

[3] Franco, J., Vasseur, O., Corre, N., Sergent, M., 2009. Minimum spanning tree: A new approach to assess the quality of the design of computer experiments. Chemom. Intell. Lab. Syst. 97 (2), 164–169.

[4] Gilquin, L., Rugama, L. A. J., Arnaud, E., Hickernell, F. J., Monod, H., Prieur, C., 2016. Iterative construction of replicated designs based on Sobol' sequences, preprint available at https://hal.inria.fr/hal-01349444.

[5] Hedayat, A. S., Sloane, N. J. A., Stufken, J., 1999. Orthogonal Arrays: Theory and Applications, new york: springer Edition. Springer Series in Statist.

[6] Hoeffding, W., 1948. A class of statistics with asymptotically normal distributions. Annals of Mathematical Statistics 19 (3), 293–325.

[7] Janon, A., Klein, T., Lagnoux, A., Nodet, M., Prieur, C., 2014. Asymptotic normality and efficiency of two Sobol' index estimators. ESAIM Probab. Stat. 18, 342–364.

[8] Jonshon, M. E., Moore, L. M., Ylvisaker, D., 1990. Minmax and maximin distance designs. J. Statist. Plann. Inference 26 (2), 131–148.

[9] Mara, T. A., Joseph, O. R., 2008. Comparison of some efficient methods to evaluate the main effect of computer model factors. Journal of Statistical Computation and Simulation 78 (2), 167–178.

[10] McKay, M. D., Conover, W. J., Beckman, R. J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21, 239–245.

[11] Monod, H., Naud, C., Makowski, D., 2006. Uncertainty and sensitivity analysis for crop models. Elsevier, Ch. 3, pp. 55–100.

[12] Morokoff, W. J., Caflisch, R. E., 1994. Quasi-random sequences and their discrepancies. SIAM J. Sci. Comput. 15 (16), 12511279.

[13] Owen, A. B., 2013. Better estimation of small Sobol' sensitivity indices. ACM Trans. Model. Comput. Simul. 23 (2), :11.

[14] Qian, P. Z. G., 2009. Nested Latin hypercube designs. Biometrika 96 (4), 957–970.

[15] Qian, P. Z. G., Ai, M., Wu, C. F. J., 2009. Construction of nested space-filling designs. Ann. Stat. 37 (6A), 3616–3643.

[16] Qian, P. Z. G., Tang, B., Wu, C. F. J., 2009. Nested space-filling designs for computer experiments with two levels of accuracy. Stat. Sinica 19, 287–300.

[17] Saltelli, A., 2002. Making best use of models evaluations to compute sensitivity indices. Comput. Phys. Commun. 145 (2), 280–297.

[18] Saltelli, A., Chan, K., Scott, E. M., 2008. Sensitivity Analysis. Wiley.

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

[20] Stinson, D. R., Massey, J. L., 1995. An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions. J. Cryptology 8, 67–173.

[21] Tissot, J. Y., Prieur, C., 2015. A randomized orthogonal array-based procedure for the estimation of first- and second-order Sobol' indices. J. Statist. Comput. Simulation 85, 1358–1381.

Acronyms and Symbols

$\subsetneq$	(strict) inclusion symbol
$\subset$	inclusion symbol
$\lvert x\rvert$	cardinality of a set x
x^T	transpose of x
Π_n	set of all the permutations on $\{1,\dots,n\}$
$OA_\lambda(q,d,t)$	Orthogonal array of index λ, levels q and strength t
$GF(q)$	Galois field of order q
$\Diamond$	operator symbol
F	cumulative distribution function
F^{-1}	quantile function
$\lvert\lvert.\rvert\rvert$	absolute value function
$\underline{S}_u$	closed Sobol' index of order I
$\widehat{\underline{S}}_u$	estimator of $\underline{S}_u$
d	input space dimension
$\mathbb{R}$	real coordinate space
$(\Omega,\mathcal{A},\mathbb{P})$	probability space
P_X	distribution function of a random variable X
$\mathbb{L}^2(P_X)$	space of square integrable functions
E	expectation symbol
Var	variance symbol
Cov	covariance symbol
$\mathbb{N}$	set of positive integers