An Efficient Approach to Deal with the Curse of Dimensionality in Sensitivity Analysis Computations

Ratto M., Saltelli A.

Institute for the Protection and Security of the Citizen (IPSC)
JRC, European Commission, TP 361, 21020 ISPRA (VA) - ITALY
marco.ratto@jrc.it

Abstract. This paper deals with computations of sensitivity indices in global sensitivity analysis. Given a model $y=f(x_1,...,x_k)$, where the k input factors x_i's are uncorrelated with one another, one can see y as the realisation of a stochastic process obtained by sampling each of the x_i's from its marginal distribution. The sensitivity indices are related to the decomposition of the variance of y into terms either due to each x_i taken singularly, as well as into terms due to the cooperative effects of more than one. When the complete decomposition is considered, the number of sensitivity indices to compute is (2^k-1), making the computational cost grow exponentially with k. This has been referred to as the curse of dimensionality and makes the complete decomposition unfeasible in most practical applications. In this paper we show that the information contained in the samples used to compute suitably defined subsets A of the (2^k-1) indices can be used to compute the complementary subsets A^* of indices, at no additional cost. This property allows reducing significantly the growth of the computational costs as k increases.

1 Introduction and state-of-the-art

Global sensitivity analysis aims to quantify the relative importance of input variables or factors in determining the value of an assigned output variable y, defined as

$$y = f(x_1,..., x_k) \tag{1}$$

A recent review of applications of this discipline can be found in [1-2]. Global sensitivity analysis is based on the variance decomposition scheme proposed by Sobol' [3], whereby the total unconditional variance of y can be decomposed as:

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + ... + V_{12...k} \tag{2}$$

where

$$V_i = V\left(E\left(Y \middle| x_i\right)\right) \tag{3}$$

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 196–205, 2002.

$$V_{ij} = V\left(E\left(Y\middle|x_i, x_j\right)\right) - V_i - V_j \tag{4}$$

and so on. The development in (2) contains k terms of the first order V_i, $k(k\text{-}1)/2$ terms of the second order V_{ij} and so on, till the last term of order k, for a total of $(2^k\text{-}1)$ terms. The V_{ij} terms are the second order (or two-way) terms, analogous to the second order effects described in experimental design textbooks (see e.g. [4]). The V_{ij} terms capture that part of the effect of x_i and x_j that is not described by the first order terms. If one divides the terms in equation (2) by the unconditional variance V, one obtains the sensitivity indices of increasing order S_i, S_{ij} and so on. The so defined sensitivity indices are nicely scaled in [0, 1]. Formula (2) has a long history, and various authors have proposed different versions of it. A discussion can be found in [5-7]. Sobol's version of formula (2) is based on a decomposition of the function f itself into terms of increasing dimensionality, i.e.:

$$f(x_1, x_2, ..., x_k) = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + ... + f_{12...k} \tag{5}$$

where each term is function only of the factors in its index, i.e. $f_i = f_i(x_i)$, $f_{ij} = f_{ij}(x_i, x_j)$ and so on. Decompositions (2, 5) are unique provided that the input factors are independent and that each individual term (5) is square integrable and the integral over any of its own variables is zero [3].

Considering a generic term of cardinality c in the development (2), the set $\mathbf{u} = (x_{i_1}, x_{i_2}, ..., x_{i_c})$ of the factors in the generic term and the remaining set $\mathbf{v} = (x_{l_1}, x_{l_2}, ..., x_{l_{k-c}})$ can be defined. The evaluation of the term requires the computation a multidimensional integral of the following form

$$V\left(E\left(y\middle|\mathbf{u}\right)\right) = \int E^2\left(y\middle|\mathbf{u} = \tilde{\mathbf{u}}\right) p(\tilde{\mathbf{u}}) d\tilde{\mathbf{u}} - E^2(y) \tag{6}$$

where $p(\mathbf{u})$ denotes the joint probability density function of factors in the set $\mathbf{u}$, which, upon orthogonality, can be expressed as

$$p(\mathbf{u})d\mathbf{u} = \prod_{j=i_1}^{i_c} \left(p_j\left(x_j\right)dx_j\right) \tag{7}$$

Equation (6) is computationally impractical. In a Monte Carlo frame, it implies a double loop: the inner one to compute the conditional expectation of y conditioned to the subset $\mathbf{u}$ (integral over the complementary $\mathbf{v}$ subspace) and the outer to compute the integral over the $\mathbf{u}$ subspace. The integral in (6) has been rewritten by Ishigami and Homma [8], Sobol' [3], as:

$$\begin{aligned}
&\int E^2\left(y\middle|\mathbf{u} = \tilde{\mathbf{u}}\right) p(\tilde{\mathbf{u}})\, d\tilde{\mathbf{u}} = \int \left\{\int f\left(\tilde{\mathbf{u}}, \mathbf{v}\right) p\left(\mathbf{v}\right) d\mathbf{v}\right\}^2 p(\tilde{\mathbf{u}})\, d\tilde{\mathbf{u}} = \\
&\int \left\{\int f\left(\tilde{\mathbf{u}}, \mathbf{v}\right) f\left(\tilde{\mathbf{u}}, \mathbf{v}'\right) p\left(\mathbf{v}\right) d\mathbf{v}\ p\left(\mathbf{v}'\right) d\mathbf{v}\right\} p(\tilde{\mathbf{u}})\, d\tilde{\mathbf{u}} = \\
&\int f\left(\mathbf{u}, \mathbf{v}\right) f\left(\mathbf{u}, \mathbf{v}'\right) p\left(\mathbf{v}\right) d\mathbf{v}\ p\left(\mathbf{v}'\right) d\mathbf{v}'\ p\left(\mathbf{u}\right) d\mathbf{u}
\end{aligned} \tag{8}$$

The expedient of using the additional integration variable primed, allows us to realise that the integral in (8) is the expectation value of the function F of a set of (2k-c) factors:

$$\begin{aligned}&F(\mathbf{u},\mathbf{v},\mathbf{v}')=f(\mathbf{u},\mathbf{v})f(\mathbf{u},\mathbf{v}')=\\&f(x_{i_1},x_{i_2},...,x_{i_c},x_{l_1},x_{l_2},...,x_{l_{k-c}})f(x_{i_1},x_{i_2},...,x_{i_c},x'_{l_1},x'_{l_2},...,x'_{l_{k-c}})=\\&f(x_1,x_2,...x_k)f(x_{i_1},x_{i_2},...,x_{i_c},x'_{l_1},x'_{l_2},...,x'_{l_{k-c}})\end{aligned}\qquad\textbf{(9)}$$

The integral (8) can be hence computed using a single Monte Carlo loop. The Monte Carlo procedure that follows was proposed by Saltelli et al. [9].

Two input sample matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are generated:

$$\mathbf{M}_1=\begin{matrix}x_{11}&x_{12}&...&x_{1k}\\x_{21}&x_{22}&...&x_{2k}\\...&&&\\x_{n1}&x_{n2}&...&x_{nk}\end{matrix}\;,\qquad\mathbf{M}_2=\begin{matrix}x'_{11}&x'_{12}&...&x'_{1k}\\x'_{21}&x'_{22}&...&x'_{2k}\\...&&&\\x'_{n1}&x'_{n2}&...&x'_{nk}\end{matrix}\qquad\textbf{(10)}$$

where n is the sample size used for the Monte Carlo estimate. In order to estimate the sensitivity measure for a generic subset **u**, i.e.

$$S_{\mathbf{u}}=V(E(y|\mathbf{u}))/V(y)\qquad\textbf{(11)}$$

$$V(E(y|\mathbf{u}))=U_{\mathbf{u}}-E^2(y)\qquad\textbf{(12)}$$

$$U_{\mathbf{u}}=\int\int\;\int f(\mathbf{u},\mathbf{v})f(\mathbf{u},\mathbf{v}')d\mathbf{u}d\mathbf{v}d\mathbf{v}'\qquad\textbf{(13)}$$

we need an estimate for both $E^2(y)$ and $U_{\mathbf{u}}$. The former can be either obtained from values of y computed on the sample in $\mathbf{M}_1$ or $\mathbf{M}_2$. $U_{\mathbf{u}}$ can be obtained from values of y computed on the matrix $\mathbf{N_u}$, whose columns are taken from matrix $\mathbf{M}_1$ for factors in subset **u** and from matrix $\mathbf{M}_2$ for factors in subset **v** (i.e. **v'**):

$$\mathbf{N_u}[1,...,n;i_1,...,i_c]=\begin{matrix}x_{1i_1}&...&x_{1i_c}\\x_{2i_1}&...&x_{2i_c}\\...&...&...\\x_{ni_1}&...&x_{ni_c}\end{matrix}\quad\mathbf{N_u}[1,...,n;l_1,...,l_{k-c}]=\begin{matrix}x'_{1l_1}&...&x'_{1l_{k-c}}\\x'_{2l_1}&...&x'_{2l_{k-c}}\\...&...&...\\x'_{nl_1}&...&x'_{nl_{k-c}}\end{matrix}\qquad\textbf{(14)}$$

i.e. by:

$$\hat{U}_{\mathbf{u}}=\frac{1}{n-1}\sum_{r=1}^{n}f(x_{r1},x_{r2},...,x_{rk})f(x_{ri_1},x_{ri_2},...,x_{ri_c},x'_{rl_1},x'_{rl_2},...,x'_{rl_{k-c}})\qquad\textbf{(15)}$$

If one thinks of matrix $\mathbf{M}_1$ as the “sample“ matrix, and of $\mathbf{M}_2$ as the “re-sample” matrix, then $\hat{U}_{\mathbf{u}}$ is obtained from products of values of f computed from the sample matrix times values of f computed from $\mathbf{N_u}$, i.e. a matrix where all factors **v** are resampled. This procedure allows estimating all terms but the last one, quantifying the k order interaction effect[1]. The latter can be estimated by difference, subtracting from the unconditional variance the remaining (2^k-2) terms. Error estimates for $\hat{U}_{\mathbf{u}}$'s are discussed in the original reference of Sobol'. A bootstrap based alternative is discussed in [5].

So, a complete analysis of all effects of increasing order for the k input factors can be obtained by computing all terms in (2), but these are as many as (2^k-1). This problem has been referred to by Rabitz et al. [7] as "the curse of dimensionality". The computational cost of estimating all effects in (2) is in fact as high as $n(2^k\text{-}1)$, where again n is the sample size used to estimate one individual effect[2].

2 A More Efficient Approach

Consider to have the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, and to have run the model for the parameter values of matrix $\mathbf{M}_1$ and for a subset m of matrices ($\mathbf{N}_{\mathbf{u}1}$, …, $\mathbf{N}_{\mathbf{u}m}$) of the whole set of (2^k-1) $\mathbf{N}_{\mathbf{u}_i}$ matrices to be considered. According to the standard procedure described in the previous section, m indices can be computed by applying the estimator (15), by multiplying one at a time the vectors of n model runs for matrices ($\mathbf{N}_{\mathbf{u}1}$, …, $\mathbf{N}_{\mathbf{u}m}$), that we define [$f(\mathbf{N}_{\mathbf{u}1})$, …, $f(\mathbf{N}_{\mathbf{u}m})$], by the vector of n model runs for parameter values $\mathbf{M}_1$, that we define $f(\mathbf{M}_1)$. However, nothing impedes to apply the same estimator (15) to all pair-wise combinations of the set of m vectors of model evaluations [$f(\mathbf{N}_{\mathbf{u}1})$, …, $f(\mathbf{N}_{\mathbf{u}m})$]. By doing so, another set of $m(m\text{-}1)/2$ of $\hat{U}_{\mathbf{u}}$ estimates, *with repetitions*, can be obtained. Such an additional set has no additional computational cost, and so can be used for making the computation of the whole set of (2^k-1) indices more efficient. This idea has been proposed by Saltelli [10], who demonstrated that, once first order effects and total effects have been computed, at the cost of $n(k+2)$ model runs, double estimates can be obtained for the effects of all groups of two factors taken together and all groups of (k-2) factors taken together, at no additional cost[3].

Some formalism is now needed, in order to allow a correct problem formulation for the efficient approach. The matrices $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{N}_{\mathbf{u}_i}$ can be represented by k-dimensional Boolean arrays as follows:

$$a_{(1,\dots,k)} = [0,0,\dots,0] \text{ for } \mathbf{M}_1; a_0 = [1,1,\dots,1] \text{ for } \mathbf{M}_2$$
$$a_{(i_1,\dots,i_{c_i})}[j] = \begin{cases} 0 \text{ if } j \in [i_1,\dots,i_{c_i}] \\ 1 \text{ if } j \in [l_1,\dots,l_{k-c_i}] \end{cases} \text{ for } \mathbf{N}_{\mathbf{u}_i} \tag{16}$$

[1] It is easy to see that for the k order term, the matrix $\mathbf{N}_{\mathbf{u}}$ equals the sample matrix and the estimator (15) would simply give the unconditional variance.

[2] $n(2^k\text{-}2)$ is needed to compute all effects but the k order, and n more to compute $\hat{E}(y)$, $V(y)$.

[3] Total indices are defined as S_{Ti}=1-$S_{\sim i}$, where ~i means all factors except i, and quantifies the effect of i alone (main effect) plus the interaction terms with all other factors [1,3,10].

where c_i is the cardinality of the element $\mathbf{u}_i$ of the decomposition (2) and $(l_1,\ldots,l_{k\text{-}c_i})$ is., as usual, the array of indices of the complementary set of factors $\mathbf{v}$.

For example, for k=3, the whole set of arrays is defined by:

$$\begin{array}{lll} a_{123}=(0,0,0) & a_1=(0,1,1) & a_{12}=(0,0,1) \\ & a_2=(1,0,1) & a_{13}=(0,1,0) \\ a_0=(1,1,1) & a_3=(1,1,0) & a_{23}=(1,0,0) \end{array} \qquad (\mathbf{17})$$

and the estimates $\hat{U}_{\mathbf{u}}$ and their Boolean representations s_i can be expressed by:

$$\begin{aligned}
\hat{U}_1 &= 1/(n-1)f(\mathbf{M}_1)\cdot f(\mathbf{N}_1);\ s_1=(1,0,0)=(0,0,0)\otimes(0,1,1)\\
\hat{U}_2 &= 1/(n-1)f(\mathbf{M}_1)\cdot f(\mathbf{N}_2);\ s_1=(0,1,0)=(0,0,0)\otimes(1,0,1)\\
\hat{U}_3 &= 1/(n-1)f(\mathbf{M}_1)\cdot f(\mathbf{N}_3);\ s_1=(0,0,1)=(0,0,0)\otimes(1,1,0)\\
\hat{U}_{12} &= 1/(n-1)f(\mathbf{M}_1)f(\mathbf{N}_{12});\ s_{12}=(1,1,0)=(0,0,0)\otimes(0,0,1)\\
\hat{U}_{13} &= 1/(n-1)f(\mathbf{M}_1)f(\mathbf{N}_{13});\ s_{13}=(1,0,1)=(0,0,0)\otimes(0,1,0)\\
\hat{U}_{23} &= 1/(n-1)f(\mathbf{M}_1)f(\mathbf{N}_{23});\ s_{23}=(0,1,1)=(0,0,0)\otimes(1,0,0)
\end{aligned} \qquad (\mathbf{18})$$

where $\otimes$ is a bit-wise Boolean operator which gives 1 if the elements in the array are equal and 0 if they are different. The $\otimes$ operator allows to know which estimate $\hat{U}_{\mathbf{u}}$ can be obtained by taking pair-wise combinations of the arrays of Monte Carlo runs $[f(\mathbf{N}_{\mathbf{u}1}),\ldots,f(\mathbf{N}_{\mathbf{u}m})]$. For example:

$$a_{13}\otimes a_3=(0,1,0)\otimes(1,1,0)=(0,1,1)=s_{23}\Rightarrow 1/(n-1)f(\mathbf{N}_{13})\cdot f(\mathbf{N}_3)=\hat{U}_{23} \qquad (\mathbf{19})$$

With this formalism, the various array elements a_i can also be interpreted as co-ordinates of the vertices of a k-dimensional unit hypercube, while the sensitivity indices can be represented by edges and hyper-diagonals joining couples of vertices a_i. With this interpretation the standard procedure consists of drawing all edges and hyper-diagonals starting from the origin a_{123}.

Another interpretation of arrays a_i is that they can be seen as the k-bit binary representations of the sequence of natural numbers $[0, 1, \ldots, 2^k\text{-}1]\in\aleph$. This can be useful for a synthetic representation of arrays a_i trough integer numbers.

Theorem 1

Let m^* be the minimum dimension of the subset A of *opportunely* chosen elements $[a_{\mathbf{u}1},\ldots,a_{\mathbf{u}m^*}]$, for which, combining pair-wise the elements of A through the $\otimes$ operator, the whole set of s_i elements can be obtained, then

$$m_{MIN}=\mathrm{int}^+\left[\left(1+\sqrt{1+8(2^k-2)}\right)\Big/2\right]\le m^*<2^k-2 \qquad (\mathbf{20})$$

where $\mathrm{int}^+(z)$ is the smallest integer larger the real number z.

Proof

First let us demonstrate the right-hand side inequality. The cost of the standard procedure is as high as $n(2^k\text{-}1)^2$. However, for efficiency purposes, $\hat{E}(y)$ and $V(y)$ can be estimated also from any of the block matrices used for the $(2^k\text{-}2)$ $\hat{U}_{\mathbf{u}}$ estimates, making the maximum cost to be as high as $n(2^k\text{-}2)$. Moreover, the minimum number m^* is surely *smaller* than $(2^k\text{-}2)$, because by combining an arbitrary pair of arrays $[f(\mathbf{N}_{\mathbf{u}1}), \ldots, f(\mathbf{N}_{\mathbf{u}m})]$, $m{>}1$, an additional estimate $\hat{U}_{\mathbf{u}(m+1)}$ can be computed at zero cost, reducing the total cost *at least* by n.

The left-hand side can be demonstrated by introducing the constraint that the number of pair-wise combinations of the m^* elements must be at least as high as $(2^k\text{-}2)$, i.e.:

$$m^*(m^*-1)/2 \geq 2^k - 2 \tag{21}$$

which gives the minimum constraint for m^*.

Remarks

1. Inequality (21) would become and equation if each combination gives a different index, i.e. no repeated estimates are obtained in the optimal a_i subset.
2. Theorem 1 shows that the order of the growth rate of the computational cost of the full decomposition can be halved, since the cost is reduced from $o(2^k)$ to $o(2^{(k+1)/2})$. This makes the complete decomposition feasible at least for 'not too high' values of k (e.g. $k \leq 11$, see next sections). On the other hand the exponential form is unaltered. This implies that for large values of k, the curse of dimensionality problem remains impracticable.
3. Theorem 1 does not tell us anything about the choice of the m^* elements allowing the efficient approach. This will be the solution of an optimisation problem, which we have not yet solved for the general case, but only for some values of k. In section 3 we will show a geometrical solution of the optimisation problem for k=3, 4. In section 4 we will show some experimental results obtained from a simple (but not optimal!) algorithm for the selection of suitable elements, which allows appreciating the degree of simplification offered by the proposed approach for larger k values.

2.1 Geometrical Solution of the Optimisation Problem for *k*=3, 4

For the case of three factors, formula (20) gives that m_{MIN}=4. The complete decomposition in this case reads:

$$V(y) = V_1 + V_2 + V_3 + V_{12} + V_{13} + V_{23} + V_{123} \tag{22}$$

Let us consider the 3D unit-cube representing our optimisation problem. The three main effects can be represented by choosing three face diagonals taken on three distinct non-parallel faces. The three second order effects are represented by choosing three non-parallel edges. Our optimisation problem can be seen as identifying the minimum set of vertices, which allow drawing three non-parallel edges and three non-parallel faces. It is easy to verify that it is sufficient to consider the 4 vertices shown in Fig. 1 in order to draw all required edges and diagonals. Since in this case $m^*=m_{MIN}$, the optimisation problem for k=3 is solved.

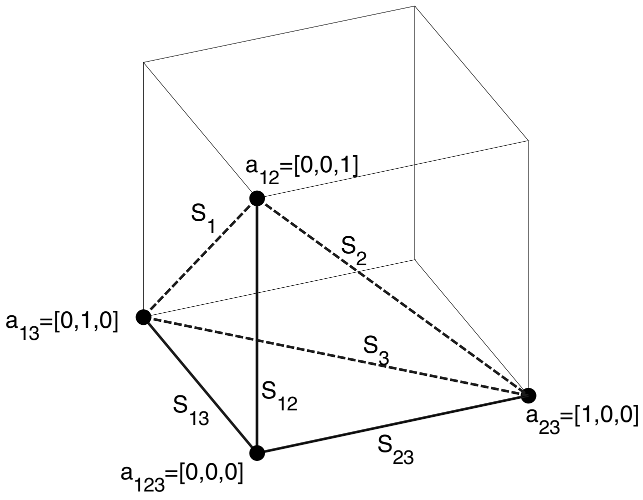

Fig. 1. Graphical interpretation of the efficient approach for k=3. Solid lines are second order effects [edges]; dashed lines are first order effects [face diagonals].

Table 1. Results of the application of the operator ⊗ to all pair-wise combinations of optimal subset of elements a_i for k=4.

Bits ↔ ℵ		a_{1234}	a_{123}	a_{124}	a_{134}	a_{234}	a_0
[0,0,0,0] ↔ 0	a_{1234}	$\hat{V}(y)$					
[0,0,0,1] ↔ 1	a_{123}	$\hat{U}_{123}$	$\hat{V}(y)$				
[0,0,1,0] ↔ 2	a_{124}	$\hat{U}_{124}$	$\hat{U}_{12}$	$\hat{V}(y)$			
[0,1,0,0] ↔ 4	a_{134}	$\hat{U}_{134}$	$\hat{U}_{13}$	$\hat{U}_{14}$	$\hat{V}(y)$		
[1,0,0,0] ↔ 8	a_{234}	$\hat{U}_{234}$	$\hat{U}_{23}$	$\hat{U}_{24}$	$\hat{U}_{34}$	$\hat{V}(y)$	
[1,1,1,1] ↔ 15	a_0	$\hat{E}^2(y)$	$\hat{U}_4$	$\hat{U}_3$	$\hat{U}_2$	$\hat{U}_1$	$\hat{V}(y)$

In the case of k=4, formula (20) gives that m_{MIN}=6. By extending the geometrical solution found for k=3, we took as initial trial subset of vertices the origin a_{1234} and its 4 adjacent vertices: a_{234}, a_{134}, a_{124}, a_{123}. Finally by adding the vertex opposite to the origin a_0, it can be easily verified from Table 1 that the whole set of 15 indices can be computed from these 6 elements. Again, since our estimated m^* equals m_{MIN}, the solution find is the optimal solution for the efficient approach. In the first column of Table 1, the 4-bit binary representation of a_i and the natural numbers corresponding to such a binary array are also shown. A final remark regards the cross-combination of two complementary elements (see a_{1234} and a_0 in Table 1). In such a case, the result of the operator ⊗ gives the squared expected value, since we are making the scalar product of two independent random vectors. On the other hand, applying the operator ⊗ to repeated elements (diagonal elements of the matrix), we get estimates of the unconditional variance of y.

2.2 Some Experimental Results for k>4

For k>4, we were not able to find similar geometrical considerations to find the optimal subset of vertices in the k dimensional hyper-cube. The problem should be solved with an optimisation algorithm. Such an optimisation problem can be a formidable task. Consider the easiest case of k=5. The total number of elements among which the optimal subset has to be found is as large as 2^5 = 32, while m_{MIN}=9. If no optimisation algorithm is applied, but a complete search of combinations is performed, the combinatory study requires the evaluation of 32 elements grouped by 9, i.e. 28048800 possible combinations. Further examples of costs for the full search procedure are shown in Table 2. It can be seen that, for k>5, a procedure based on a full search of combinations is clearly unaffordable, at least on a workstation or PC.

Table 2. Cost of a plain search of combinations for some values of k.

k	m_{MIN}	Elements (2^k)	Combinations for $m^*=m_{MIN}$	Combinations for $m^*=m_{MIN}+1$	Combinations for $m^*=m_{MIN}+2$
5	9	32	28048800	64512240	-
6	12	64	$3.3\ 10^{12}$	$1.3\ 10^{13}$	$4.8\ 10^{13}$
7	17	128	$6.1\ 10^{20}$	$3.8\ 10^{21}$	$2.2\ 10^{22}$

So, to find possible suitable combinations for k>4, we applied a simple and relatively quick algorithm, which is not an optimisation algorithm but gives sufficiently small values of m^* to appreciate the advantage of the new approach. The algorithm takes an initial trial subset of k+1 elements, given by the origin $a_{1,\ldots,k}$ and the k adjacent vertices, as we did for k=3, 4. Then it adds one element at a time in such a way that the additional number of indices computable introducing the new element is maximum. The algorithm stops when the elements a_i selected are sufficient to compute the whole set of (2^k-1) indices. In the case of k=5, the algorithm gave a number of elements m^*=10. In such a case, we verified with a complete search of combinations that none of the 28048800 combinations of 9 elements allows the computation of the whole set of indices, so that also for k=5 the optimisation problem was solved.

The results of this procedure for k=3, …, 11 are shown in Table 3. The advantages of the new approach are clear. If the standard procedure begins to be hard to implement for k=7, with the new approach, we can afford a complete analysis at least for k=11. Moreover, even if the 'true' values of m^* were smaller than the estimates $\hat{m}^*$ shown in Table 3, the subsets of a_i elements that we found were sufficiently small to bridge most of the gap between the cost of the standard procedure and the limit *m(MIN)*. In fact, given the latter inferior limit, the further improvement, which may be given by the true optimisation, will not be as large as the one presented here.

In Table 4 we show the subsets of a_i elements represented by arrays of natural numbers in the interval [0, 2^k-1]. By taking the k-bit binary representation of such arrays, we get the a_i elements allowing the estimation of all terms in the decomposition (2).

Table 3. Comparison of the computational cost of the standard and the new procedure for some values of k.

k	Number of indices (2^k-1)	Cost of the standard procedure	Estimated cost of the new procedure $\hat{m}^*$	Minimum limit for cost [formula (20)]
3	7	7 n	4 n[4]	4 n
4	15	15 n	6 n[4]	6 n
5	31	31 n	10 n[4]	9 n
6	63	63 n	14 n	12 n
7	127	127 n	20 n	17 n
8	255	255 n	29 n	24 n
9	511	511 n	42 n	33 n
10	1023	1023 n	67 n	46 n
11	2047	2047n	98 n	65 n

Table 4. Arrays of natural numbers representing the optimal a_i conbinations detected by the simplified algorithm for some values of k.

k	Optimal combinations for the new approach
3	[0 1 2 4]
4	[0 1 2 4 8 15]
5	[0 1 2 4 8 15 16 17 18 19]
6	[0 1 2 3 4 5 6 8 16 24 32 40 48 63]
7	[0 1 2 4 8 15 16 22 28 32 44 51 57 64 77 85 94 106 107 112]

3 Conclusions

In this paper we presented some efficient procedures for numerical experiments aimed at sensitivity analysis of model output. We have focused here on the computation of sensitivity indices that are based on decomposing the variance of the target function in a quantitative fashion. The approach presented aimed to identify to what extent it is

[4] In this case the optimisation problem has been solved, so the estimate is correct.

possible to fight the so-called "curse of dimensionality", that hinders the use of quantitative sensitivity analysis for computationally expensive models.

The approach allows reducing by a factor 2 the growth rate of computational effort required for the complete decomposition of the unconditional variance of a model output. It consists of using the information contained in the Monte Carlo samples applied for the computation of a given subset of terms, to compute the remaining set of terms, at no additional cost. The complete solution of the optimisation problem is still in progress and we managed to treat it in a simplified way only for k<12.

On the base of this incomplete results, we can state that even assuming for n the value of 1000, the new procedure can make the full decomposition affordable for models whose cost per run is in the range from milliseconds or lower to some seconds. If k does not exceed 6, the model cost per run can increase to some minutes. For models whose execution is in the tenths of minutes to a day range, quantitative methods are not applicable and efficient qualitative methods such as that of Morris [11] should be used (see [12] for a review).

References

1. Saltelli, A., Tarantola, S., Campolongo, F.: Sensitivity analysis as an ingredient of modelling. Statistical Science 15 (2000) 377-395.
2. Saltelli, A., Chan, K., Scott, M. (eds.): Sensitivity Analysis. Wiley Series in Probability and Statistics. John Wiley and Sons, Chichester (2000).
3. Sobol', I. M.: Sensitivity estimates for nonlinear mathematical models. Matematicheskoe Modelirovanie 2 (1990) 112-118, translated as: Sobol', I. M.: Sensitivity analysis for nonlinear mathematical models. Mathematical Modelling & Computational Experiment 1 (1993) 407-414.
4. Box, G. E. P., Hunter, W. G., Hunter, J. S.: Statistics for experimenters. John Wiley and Sons, New York (1978).
5. Archer, G., Saltelli, A., Sobol', I. M.: Sensitivity measures, ANOVA like techniques and the use of bootstrap. Journal of Statistical Computation and Simulation 58 (1997) 99-120.
6. Rabitz, H., Aliş, Ö. F., Shorter, J., Shim, K.: Efficient input-output representations. Computer Physics Communications 117 (1999) 11-20.
7. Rabitz, H., Aliş, Ö. F.: Managing the Tyranny of Parameters in Mathematical Modelling of Physical Systems. In: Saltelli, A. Chan K., Scott M. (eds.): Sensitivity Analysis. Wiley Series in Probability and Statistics. John Wiley and Sons, Chichester (2000) 199-223
8. Ishigami, T., Homma, T.: An importance quantification technique in uncertainty analysis for computer models. Proceedings of the ISUMA '90, First International Symposium on Uncertainty Modelling and Analysis, University of Maryland, December 3-6 (1990).
9. Saltelli A., Andres, T. H., Homma, T.: Some new techniques in sensitivity analysis of model output. Computational Statistics and Data Analysis 15 (1993) 211-238
10. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. Submitted to Computer Physics Communications (2002).
11. Morris, M. D.: Factorial Sampling Plans for Preliminary Computational Experiments. Technometrics 33 (1991) 161-174
12. Campolongo, F., Saltelli, A.: Design of experiment. In: Saltelli, A. Chan K., Scott M. (eds.): Sensitivity Analysis. Wiley Series in Probability and Statistics. John Wiley and Sons, Chichester (2000) 51-63.