

AN EFFICIENT DESIGN OF EXPERIMENTS USING RBD-FAST

SÖREN WENZEL¹, ELENA SLOMSKI-VETTER¹ AND TOBIAS MELZ¹

¹Technical University of Darmstadt,
Otto-Bernd-Str. 2, 64287 Darmstadt,
soeren.wenzel@sam.tu-darmstadt.de

Key words: spectral analysis, sensitivity analysis, RBD-FAST, HFR, input parameters, brute force algorithm.

Abstract. This research work deals with the efficient Design of Experiments (DoE) for the hybrid variant of the sensitivity analysis method RBD-FAST called Hybrid FAST RBD (HFR). RBD-FAST is a combination of the Random Balance Design (RBD) and the Fourier Amplitude Sensitivity analysis Test (FAST). Commonly, sensitivity analyses are used to determine whether input parameters have an influence on a target value or not. Currently, there is only little research to be found for the HFR method. The HFR method separates the input parameters into groups. Different constraints must be met for an optimal grouping. Theoretically, for every square number an optimal grouping exists, but only for squares of primes an optimal grouping is known to exist. An experiment with any number of input parameters needs as many samples as an experiment with the next higher square of a prime number as the number of input parameters. In this research work optimal groupings for squares of non-primes for the HFR method are found using a brute force algorithm.

1 INTRODUCTION

A key aspect of science is to test hypothesis with the aim to confirm or deny a hypothesis. Hypothesis can be tested by analysing experiments and experimental data. While trivial experiments can easily be analysed, non-trivial experiments using numerical simulation models can be analysed using a sensitivity analysis method (Iooss B. et. al. [1]). For an efficient sensitivity analysis, as few experiments as possible should be required without degrading the quality of the output. If it is possible to use fewer trials for a sensitivity analysis, while keeping the same evidence, there is a real improvement in sensitivity analysis. In this research work, an improvement of the HFR method by new group formation is presented.

In FAST, experiments are performed in such a way that the effects can be calculated using a Fourier transformation. Every input parameter is associated with a unique frequency (Cukier et al. [2]). All input parameters can be transformed using the unique frequency in a one-dimensional space while the multi-dimensional space of input parameters is fully covered. With a multi-dimensional Fourier transformation main effects and interaction effects of input parameters can be calculated. In addition, total Sobol indices can be calculated using FAST (Saltelli et al. [3]). The chosen frequencies and transformations influence the results directly (Saltelli et al. [4]). For more than 10 input parameters the result will be biased. (Tissot et al. [5])

If a lot of input parameters are to be observed, the one-dimensional space must be sampled more often. The number of samples increases linearly with the square root of the number of input parameters (Saltelli [6]).

In contrast to FAST, RBD-FAST uses one frequency, although it is based on FAST (Tarantola et al. [7]). With RBD-FAST, multi-dimensional space of input parameters can be partially sampled. To do this, the values of the input parameters are randomly permuted. To calculate the main effect, the output values will be sorted by one input parameter at a time. The interaction effects cannot be calculated, but it is comparatively easy to choose a frequency, as there are mostly no constraints and 1 Hz can be used as a frequency (Mara et al. [8]).

Within HFR, input parameters are divided into groups with most equal cardinality. Within one group, the FAST algorithm is used, which means each input parameter is associated with a unique frequency within a group. But between the groups, RBD-FAST is used because several input parameters have the same frequency. HFR combines the calculation effort advantage of RBD-FAST with the calculation of first-order interaction effects of FAST. HFR benefits from optimal grouping of the input parameters, so that less experiments are needed for the analysis while maintaining the same evidence. Tissot [5] describes a specific optimal grouping for specific numbers of input parameters. This research work tries to find more optimal groupings, that are yet to be documented in science. It describes how to find valid groupings, the reliability of results that can be expected and the computation time needed.

2 EFFICIENT DESIGN OF EXPERIMENTS USING RBD-FAST

As already stated, in HFR, all the input parameters are separated into groups. Within these groups each input parameter is allocated to a unique frequency without interferences to a certain degree. If two input parameters are in the same group, the first-order interaction effects can be estimated. To estimate all first-order interaction effects, the input parameters have to be allocated to different groups. This is done via multiple experimental designs. All the experimental designs are called grouping within this research work. A grouping is valid and optimal, if it follows three rules stated by Tissot [5]:

1. Each of the experimental designs consist of q groups with q input parameters.
2. Each group in a new experimental design should contain one input parameter from each group in the first experimental design.
3. If two input parameters were already in the same group, then it is not allowed to define a group with both input parameters.

It can be seen, that by meeting the first and the third requirement, the second requirement cannot be violated. But the second requirement can be used to simplify the complexity of the algorithm to find valid groupings. Tissot [5] stated the required number of experimental designs as follows, with p as the number of input parameters and q always as a prime:

$$\begin{aligned}
 & 1 + \min q && (1) \\
 & \sqrt{p} < q && \text{for } p > 6 \\
 & q \text{ prime} \\
 & 3 && \text{for } p = 4 \text{ or } 5 \\
 & 1 && \text{for } p \leq 3
 \end{aligned}$$

A valid grouping with three experimental designs and five input parameters cannot be achieved without violating one of the requirements. In some cases, it might be more economical to violate one of the requirements but still be able to ensure the reliability of the results. In order to make it comprehensible, we will not consider this special case in the following. The required number of experimental designs is generalized and simplified as follows:

$$\begin{aligned}
 & 1 + \min q && (2) \\
 & \sqrt{p} \leq q && \text{for } p \geq 1 \\
 & q \text{ prime } \vee q = 1
 \end{aligned}$$

In the following table the number of input parameters is compared to q . $q + 1$ is the number of necessary experimental designs. The number of necessary frequencies are equal to q .

Table 1: number of input parameters associated with q , with q as a prime.

Number of input parameters	q
1	1
2-4	2
5-9	3
10-25	5
26-49	7
50-121	11
122-169	13
170-289	17
...	...

As expected all q are primes and therefore larger spacing occur between the values of q dependent on the number of input parameters. For example from 9 to 10 input parameters and from 49 to 50 input parameters the step in q is 2 and 4, respectively.

The number of necessary samples increases linearly with q , as more experimental designs are needed and within an experimental design the number of samples necessary for FAST increases with q input parameters. The aim for the value of q is to be as small as possible to be efficient. Tissot [5] only shows a solution for q as primes, in this research work, more solutions are searched using a brute force algorithm.

The grouping can easily be represented as a 3-dimensional matrix. The first index represents the position within a group. The second index represents the group number in a given experimental design and the last index is defined as the number of the experimental design. So, all the groups within an experimental design and all experimental designs within a grouping can be written in a $q \times q \times (q + 1)$ -matrix. The analytic solution for all q , which are primes, is given in such a matrix G with $i, j, k \in \{1, 2, \dots, q\}$ as:

$$G_{i,j,1} = i + (j - 1)q \quad (3)$$

$$G_{i,j,k+1} = [j - 1 + (i - 1)kq] \bmod q^2 + 1 \quad (4)$$

To obtain the experimental designs for non-squares of q , all the excess input variables should be replaced with dummy variables and should be ignored in the end, as stated by Tissot [5].

2.1 Transformations of valid groupings and shortcuts for computing

An algorithm that tests all the combinations should be complex in the order of $(q^2!)^q$. Every input parameter is represented once in every experimental design. q^2 input parameters are used for the optimal experimental design. In the first position all input parameters can be entered, in the second position all input parameters except for the one already in the first position and so on. So, for a single experimental design $q^2!$ possible combinations exist. With $q + 1$ experimental designs there are $(q^2!)^{q+1}$ possible combinations. With the help of mathematical transformations, the complexity can be reduced.

There are mathematical transformations that help to obtain a valid grouping and to randomize the outcome and to avoid bias. First of all, for all input parameters x_l with $l \in \{1, 2, \dots, q^2\}$ the indices can be switched over the whole valid grouping. For example, the input parameter x_4 can be defined as x_5 , if x_5 is defined as x_4 . This means, the first experimental design $G_{i,j,k}$ for every q can be set without loss of generality as follows:

$$G_{i,j,1} = i + (j - 1)q \quad (5)$$

$$G_{i,j,1} = \begin{bmatrix} 1 & 2 & \dots & q \\ q + 1 & q + 2 & \dots & 2q \\ \vdots & \vdots & \ddots & \vdots \\ (q - 1)q + 1 & (q - 1)q + 2 & \dots & q^2 \end{bmatrix} \quad (6)$$

The second mathematical transformation permutes the order of input variables within a group randomly, the groups within an experimental design, and the experimental designs inside a valid grouping. These transformations can transform every experimental design in a way, that the first element $G_{1,1,k}$ in every experimental design is always 1. Every element from group 1 in $G_{i,1,1}$ can be put to the first position, so that every other experimental design but the first one has the following structure, with question marks marking the unknown input parameters:

$$G_{i,j,1} = \begin{bmatrix} 1 & ? & \dots & ? \\ 2 & ? & \dots & ? \\ \vdots & \vdots & \ddots & \vdots \\ q & ? & \dots & ? \end{bmatrix} \quad (7)$$

With these simplifications, the possible groupings are decreased to $[q(q - 1)!]^q$ groupings. By exploiting the second requirement, each input parameter at position i in any group can be defined to be part of the i -th group in the first experimental design, still without loss of generality. This means, there are only q possible input parameters at any position, as there are only q input parameters in a group in the first experimental design. The number of possible groupings is therefore reduced to $q!^{q-1}$.

Another way to reduce the amount of possible grouping vastly, is to define another experimental design. Thus, the second experimental design is defined equally to the transposed first experimental design.

$$G_{i,j,2} = G_{i,j,1}^T = (i-1)q + j \quad (8)$$

$$G_{i,j,1} = \begin{bmatrix} 1 & q+1 & \cdots & q(q-1)+1 \\ 2 & q+2 & \cdots & q(q-1)+2 \\ \vdots & \vdots & \ddots & \vdots \\ q & 2q & \cdots & q^2 \end{bmatrix} \quad (9)$$

This reduces the number of possible groupings further to $q!^{q-1}q^{-1}$.

2.2 Brute force algorithm

A brute force algorithm is an algorithm which goes through all possible groupings searching for valid groupings. As the number of possible groupings grows, it is necessary to keep the calculation at a minimum, because it doubles exponentially with q . Therefore, the algorithm builds up groupings, rather than trying out every possible grouping. This means, input parameters are entered one by one and every partial grouping is validated before continuing the process.

The flowchart in Figure 1 illustrates the brute force algorithm used in the study. In the initialization, all the variables are predefined using values from chapter 2.1. The algorithm works with indexing, to remember the current position, and three matrices. The first matrix stores the actual numbers of the input parameters. The second stores pointers referring to an input parameter in a specific group in the first experimental design. Each input parameter at position i in any group is defined as part of the i -th group in the first experimental design. The progress of the algorithm is stored this way. In some cases, invalid groupings can be detected without calculating the actual number of the input parameter. The third matrix stores the information whether two input parameters are together in the same group in any experimental design or not. This information is necessary to comply rule number three. At first, all matrixes and indexes are initiated. The second step consists of entering the loop to find a valid grouping.

After the grouping is proved to be valid, it is examined for completeness. If it is incomplete, another input parameter is entered into the next position. The grouping is again tested for validity.

If the grouping is found to be invalid, the last input parameter is deleted and another one is used. But, if all input parameters are invalid at that specific position, the input parameter on the prior position is deleted and another one is used. In the end, the algorithm examines if all possible combinations have been passed.

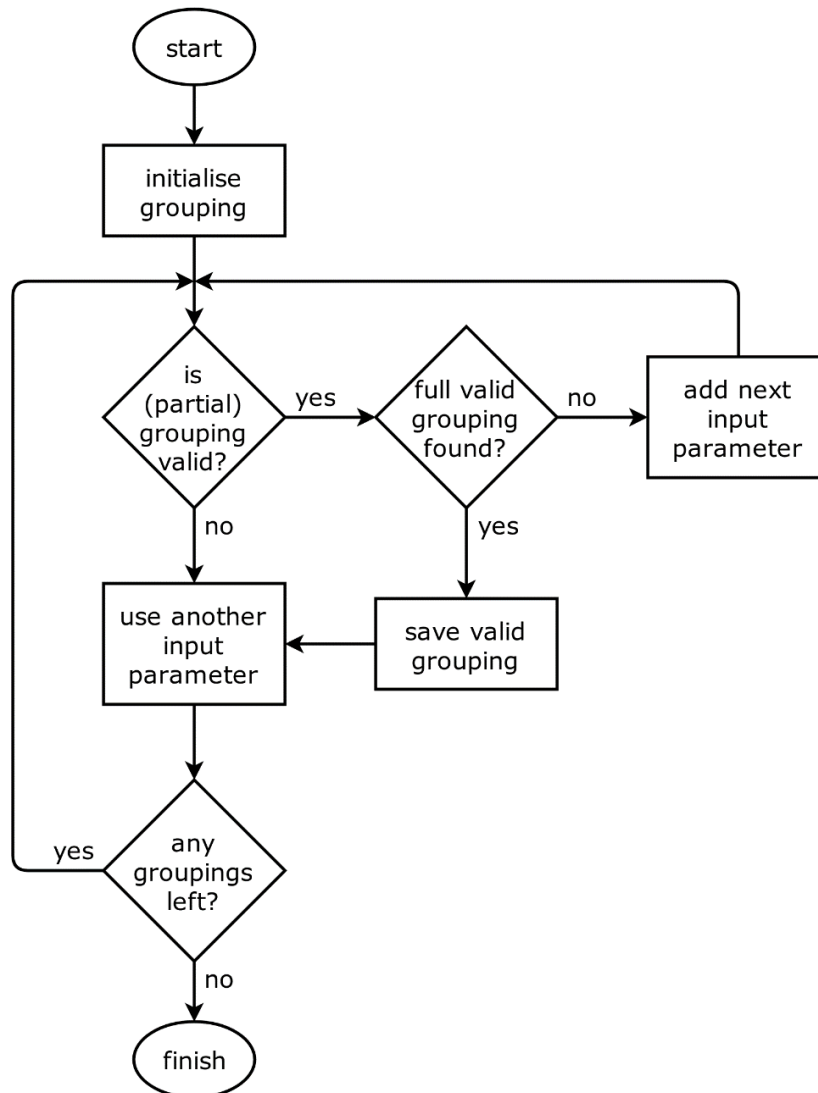


Figure 1: Simplified flowchart for the brute force algorithm to determine valid groupings.

2.3 Runtime and results

For a growing number of parameters, the brute force algorithm deals with a double exponentially growing number of possible solutions, so some values are within reach and others are not. The runtimes shown in Table 2 were measured on a conventional laptop running on a i5-3320m processor. The algorithm was executed three times and the averaged computing time was calculated.

Table 2: Runtime for several values of q .

q	runtime with one experimental design predefined	runtime with two experimental designs predefined
3	< 1s	< 1s
4	210s	< 1s
5	> 3h	5s
6	> 3h	3094s
7	> 3h	> 3h

Valid groupings were found for values of up to 5, when defined as a prime. For $5 < q \leq 20$ and q as a prime no valid groupings were found within 3 hours.

For $q = 2^n$ a multitude of valid groupings can be found. Even for $q = 16$ multiple valid groupings per second were found. For $q = 32$ no valid grouping was found within 3 hours. For all other values up to 15 of q including other square numbers no valid grouping was found. These results do not change whether one or two predefined experimental designs were used.

With one predefined experimental design, all possible valid groupings can be found in comparison to two predefined experimental designs. Here, the runtimes are a lot faster, but not all valid groupings can be found. In the case of $q = 4$ and one predefined design, valid groupings were found, which could not be transformed into valid groupings, that were found with two predefined designs.

As a conclusion the number of required experimental designs can be updated as follows:

$$\begin{aligned}
 & 1 + \min q && (10) \\
 & \sqrt{p} < q && \text{for } p > 6 \\
 & q \text{ prime } \vee q = 2^n && \\
 & 3 && \text{for } p = 4 \text{ or } 5 \\
 & 1 && \text{for } p \leq 3
 \end{aligned}$$

Table 3: input parameters associated with the number of experimental designs

No. of input parameters	No. of experimental designs
1-3	1
4; 5	3
4-9	4
10-16	5
17-25	6
26-49	8
50-64	9
65-121	12
122-169	14
170-256	17
256-289	18
...	...

The previous mentioned steps in the experimental designs between 9 and 10 input parameters and 49 and 50 are smaller than before. For 10 to 16 input parameters 5 instead of 6 experimental designs are necessary and for 50 to 64 input parameters, 9 instead of 12 experimental designs are necessary. This allows HFR to be used for 50 to 64 parameters without exceeding the limit of 10 parameters at which the underlying FAST becomes too unstable and biased.

3 CONCLUSION

A brute force algorithm was implemented to find unknown valid groupings for input parameters performing as described by Tissot [5]. It was possible to find valid groupings to have an optimal grouping with a specific number of input parameters where non optimal grouping was documented before. This means, with fewer samples the same evidence could be found. For more than 5 input parameters, the number of input parameters is square-rooted and rounded up to find q . q determines how many samples are necessary, but if no valid grouping for q is known, a larger q must be chosen. Valid groupings were only known for q as a prime. With a brute force algorithm valid groupings were found for $q = 2^n$ and $n \leq 4$.

An optimal grouping is only known, if the number of input parameters is a square of a specific q . Tissot [5] recommended breaking constraints for a specific single number of input parameters. In this way, less samples were needed for the next optimal grouping. This can be a general rule for all $q^2 + 1$ with a known optimal grouping for q^2 Input parameters. Searching for an optimal grouping for non-square input parameters might also be for future research.

The algorithm was not completed for any q without finding a valid grouping. That is, it has not been shown for any value of q that there are no valid groupings. So there is indication, but no proof, that there is a q without valid groupings.

As a conclusion, the RBD-FAST sensitivity analysis can be executed faster in some cases, dependent on the number of input parameters. As for example with 10 to 16 input parameters approximately 16.6 % less experimental designs are necessary, that directly translates to 16.6 % less samples. Also within the groups 20 % less input parameters are necessary. As groups are analyzed using FAST, another approximately 10 % less samples are needed. All in all, for 10 to 16 input parameters approximately 25 % less samples are necessary, while maintaining the

same evidence. For 50 to 64 input parameters approximately 36 % less samples are necessary. Additionally this work shows new optimal groupings for HFR, that have not yet been documented.

4 ACKNOWLEDGMENTS

We would like to thank Felix Reissner for his contribution to the algorithm.

REFERENCES

- [1] Iooss, B. and Lemaître, P. *A Review on Global Sensitivity Analysis Methods*. Uncertainty Management in Simulation-Optimization of Complex Systems (2015) **59**.
- [2] Cukier, H., Levine, R. I. and Shuler, K. *Nonlinear sensitivity analysis of multiparameter model systems*. Journal of Computational Physics (1978) **26**:1–42.
- [3] Saltelli, A., Tarantola, S., and Chan, K. P. -S. *A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output*. Technometrics (1999) **41**(1):39.
- [4] Saltelli, A. *Sensitivity analysis*. Wiley series in probability and statistics: Wiley (2004) Chichester. Reprinted.
- [5] Tissot, J. -Y. and Prieur, C. *Bias correction for the estimation of sensitivity indices based on random balance designs*. Reliability Engineering & System Safety (2012) **107**.
- [6] Saltelli, A. *Making best use of model evaluations to compute sensitivity indices*. Computer Physics Communication (2002) **145**:280–297.
- [7] Tarantola, S., Gatelli D. and Mara, T. A. *Random balance designs for the estimation of first order global sensitivity indices*. Reliability Engineering & System Safety (2006) **91**(6):717-727.
- [8] Mara, T. A. *Extension of the RBD-FAST method to the computation of global sensitivity indices*. Reliability Engineering & System Safety (2009) **94**(8).