Chemometrics and Intelligent Laboratory Systems 134 (2014) 140–147

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Repairing uniform experimental designs: Detection and/or elimination of clusters, filling gaps☆

A. Beal, J. Santiago, M. Claeys-Bruno *, M. Sergent

Aix Marseille Université, LISA EA4672, 13397, Marseille Cedex 20, France

ARTICLE INFO

Article history:
Received 26 September 2013
Received in revised form 21 March 2014
Accepted 26 March 2014
Available online 3 April 2014

Keywords:
Experimental designs
Space-Filling Designs
High dimensional space
WSP algorithm
Curvilinear Component Analysis

ABSTRACT

Construction of Space Filling Designs in high dimensional space remains difficult since powerful algorithms at low dimensions become difficult to use at higher dimensions that leads to non-uniform distribution in the factor space. We propose in this paper two approaches in order to repair designs: Curvilinear Component Analysis (CCA) and the Wootton, Sergent, Phan-Tan-Luu's algorithm called WSP in order to detect clusters and to fill gaps. Thus, CCA allows visualization of two or more very closely-spaced points in D dimensions by projecting them in a 2 dimensions space. Then identified clusters can be eliminated using the WSP algorithm. Moreover, the presence of gaps in input space could be very problematic since no information on the phenomenon is available and the WSP algorithm will be used in order to fill gaps by adding points in the “empty” zones. A new quality criterion has been proposed in order to follow the reparation steps. Examples in different dimensions are presented to illustrate these methods.

1. Introduction

In many fields, such as petrochemistry, astronomy, and meteorology, highly complex simulated models are commonly used to represent real phenomena as accurately as possible based on calculation codes. Despite real advances in processor performance, the codes simulating these phenomena still require considerable calculation times. Indeed, increasingly realistic calculations involve a large number of input variables, whose effects can be difficult to predict. It is therefore necessary to develop a strategy to determine the relevant information to supply when producing the model, such as ranking the input variables by order of importance, or having an idea of what the overall phenomenon modeled should look like. This strategy should be as effective as possible and should guarantee good quality information, even at high dimensions. Experimental designs can be used to better organize numerical simulations for this type of approach, and are currently used. However, the number of input variables – often very large (several tens, or even hundreds) – and the wide ranges of variation involved have led to standard experimental designs no longer being really appropriate. This is partly due to how they distribute points (simulations), mainly placing them at the extremities of the variables space. This is why, in numerical simulation, experimental designs known as *Space Filling Designs* (SFD) [1–3], or uniform designs, have become more popular as they distribute the points uniformly throughout the input variables space. However, not all SFD designs are equivalent in terms of the quality criteria reflecting the uniformity of point distribution, such as the intrinsic criteria *Mindist* [4–6] and *Coverage* [7]. *Mindist* is defined as the smallest Euclidian distance between two points. *Coverage* quantifies the homogeneity of spread of points and can be considered as a standard deviation of minimal distances. These criteria allow the comparison of several designs built in the same dimension with the same number of points. The design with the better quality regarding the uniform repartition and the fill-up of the space is characterized by the lowest value of *Coverage* and the highest value of *Mindist*.

In addition, many algorithms which are powerful at low dimensions ($D < 10$) become difficult to use at higher dimensions ($D > 20$ or 30). Thus, low-discrepancy sequences [8–12], such as Faure sequences, present very poor uniformity criteria at high dimensions, with low *Mindist* and high *Coverage* values. The poor conditioning of these experimental designs leads to non-uniform distribution of points throughout the space, causing the appearance of clusters and/or gaps.

Poor conditioning, in terms of non-uniform distribution, can also result from a projection of an experimental design into the sub-space of influential variables revealed by sensitivity analysis. Indeed, after sensitivity analysis, it can be useful to extract the sub-group of factors identified as influential for closer study (modeling) of the phenomenon. This involves keeping the previously performed tests (lines of the design) and only considering the columns representing influential factors. This reduction of the space is known as “folding” and can lead to the appearance of clusters or gaps in the new space.

The aim of this study was to develop a method to repair designs where points are not uniformly distributed throughout the factor space, either because of poor construction or due to folding of the initial

☆ Selected Paper from the 8th Colloquium Chemiometricum Mediterraneum (CCM VIII 2013), Bevagna, Italy, 30th June–4th July 2013.

* Corresponding author. Tel.: +33 491288186.
E-mail address: m.claeys-bruno@univ-amu.fr (M. Claeys-Bruno).

http://dx.doi.org/10.1016/j.chemolab.2014.03.017
0169-7439/© 2014 Elsevier B.V. All rights reserved.

space. To do this, we used Curvilinear Component Analysis (CCA) [13,14] to visualize clusters. Then designs were repaired using the Wooton, Sergent, Phan-Tan-Luu's selection algorithm (WSP) [15–20] to eliminate any clusters identified and to fill gaps, which strongly penalize the modeling steps. Examples of applications with 2, 8 and 20 dimensions are presented to illustrate these methods.

2. Methods

The methods presented here meet the two objectives presented, i.e., detect the presence of clusters of points in the experimental space and eliminate these clusters if necessary while also filling any gaps. We will present the principles and algorithms for these methods followed by examples of their application.

2.1. WSP algorithm

2.1.1. Algorithm

The WSP algorithm [15–20] allows uniform designs to be rapidly constructed with very good quality criteria, like *Mindist* and *Coverage*. In the WSP selection algorithm, the defined multidimensional parameter space is filled with points selected from a set of candidate points based on a preset minimal distance (d_{min}) from every other point already included in the design.

The algorithm can be summarized as follows:

Step 1 generate a set of N candidate points
Step 2 calculate the distances (D_{ij}) matrix for the N points
Step 3 choose an initial point O and a distance d_{min}
Step 4 eliminate the points I for which: $D_{\text{OI}} < d_{min}$. Point O is eliminated from the set of candidate points and will belong to the final subset
Step 5 point O is replaced by the nearest point among the remaining points
Step 6 repeat steps 4 and 5 until there are no more points to choose.

A previous study has shown [18] that the type of the initial candidate design (such as a random design, Latin Hypercubes [21–26], low discrepancy sequences [8–12] and Strauss design [27]) has no importance but only if the number of points is sufficient. The number of candidate points depends on the number of required points in the final design. Santiago et al. [18] advise to consider a number of candidate points equal to at least 5 to 10 times the final set.

Usually the initial point O is chosen as the nearest point of the center of variable space. However, if the candidate design contains a large number of points, whatever the initial point results are identical.

The number of points in the final subset depends on the value of d_{min}. If the d_{min} value increases then the number of points in the final subset decreases. The d_{min} value is determined by iteration until the number of points desired in the final subset is obtained.

Since previous studies [18] have shown that the WSP algorithm leads to uniform designs with good criteria (*Mindist* and *Coverage*) we have chosen to consider this design as presented below.

2.1.2. Reference design

We propose to use a reference design to compare the quality of any designs that could present clusters of points or gaps.

A reference design is constructed with the same dimension and the same number of points to the design to be assessed. The intrinsic uniformity criteria for this design are calculated, and the d_{min} value (equal to the *Mindist* criterion) is used to determine the shortest distance between two points. We then consider that two points separated by a distance shorter than the d_{min} *value* are closer and will form a cluster. If all the points are separated by this d_{min} value, then the spread of points is uniform.

2.1.3. Using the WSP algorithm to detect clusters

Cluster elimination consists in the suppression of points which are closely-spaced in the variable space. It appeared logical to use the WSP selection algorithm for this since this algorithm is based on calculation of distances. The difficulty lies in choosing the d_{min} value which will determine the distance from which a cluster is defined. The d_{min} value will be chosen according to the intrinsic uniformity criteria of a reference design constructed from the same conditions in number of points and dimensions. The *Mindist* is the smallest distance between two points and if we assign this value to the d_{min} then two points separated by a shorter distance than d_{min} are considered as close and will form a cluster.

2.1.4. Using the WSP algorithm to fill gaps

The absence of points in some zones of the space can be problematic as it indicates that no information on the phenomenon is available in this part of the space. The WSP algorithm can be used to fill these gaps. However, this algorithm, which constructs uniform experimental designs, is a selection algorithm retaining a set of points from a set of candidate points. It therefore cannot be used to add points. To overcome this, we concatenated two experimental designs: the one with gaps made up of "protected" points, and a second design containing a very large number of candidate points. The WSP algorithm can then be applied (with a value of d_{min} calculated from the *Mindist* criterion of the reference design) to select points from the sum of these two designs, progressively filling the gaps while retaining the protected points.

2.2. Curvilinear Component Analysis (CCA)

2.2.1. Algorithm

The aim of CCA [13,14] is to reproduce the topology of an initial space of dimension D in a smaller space of dimension p onto which we wish to project all the data. As the overall topology cannot be reproduced, CCA tries to conserve the local topology. To do this, we consider N neurons for which the input vectors $\{x_i;\ i = 1, \ldots, N\}$ in D dimensions quantify the input distribution, and for which the output vectors $\{y_i;\ i = 1, \ldots, N\}$ in p dimensions (where $p < D$) should copy the topology of x_i (Fig. 1). To do this, we use the distances between the x_i: $X_{ij} = d(x_i, x_j)$ where d is the Euclidean distance, and the corresponding output distances are: $Y_{ij} = d(y_i, y_j)$.

During projection, the objective is to make the Y_{ij} distances equivalent to the X_{ij} distances. To do this, we minimize the E_{CCA} criterion (Eq. (1)) characterizing the topological differences between the initial space and the projected space.

$$E_{CCA} = \frac{1}{2}\sum_i \sum_{i \neq j} \left(X_{ij} - Y_{ij}\right)^2 F_\lambda\left(Y_{ij}\right) \tag{1}$$

with $F_\lambda(Y_{ij}) : \mathbb{R}_+ \rightarrow [0, 1]$ a monotone decreasing function of Y_{ij}. This favors local conservation of topology. The $F_\lambda\,(Y_{ij})$ function is known as the weighting function or function of cost. Demartines and Hérault (1997) [14] first suggested taking function F with parameter λ, known as the critical distance or the neighborhood radius (Fig. 2).

The gradient descent (Eq. (2)) could be used to minimize the E_{CCA} criterion:

$$\Delta y_i = \propto \sum_{j \neq i} \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left[2 F_\lambda\left(Y_{ij}\right) - \left(X_{ij} - Y_{ij}\right) F'_\lambda\left(Y_{ij}\right)\right]\left(y_i - y_j\right) \tag{2}$$

with α is the adaptation factor.

However this adaptation rule suffers of several drawbacks. Only one neuron is adapted at a time; thus the adaptation of all neurons is heavy and the adaptation rule can fall into local minima.

Instead of moving one vector y_i according to the sum of contributions of all y_j, the CCA algorithm proposes to fix randomly a point y_i

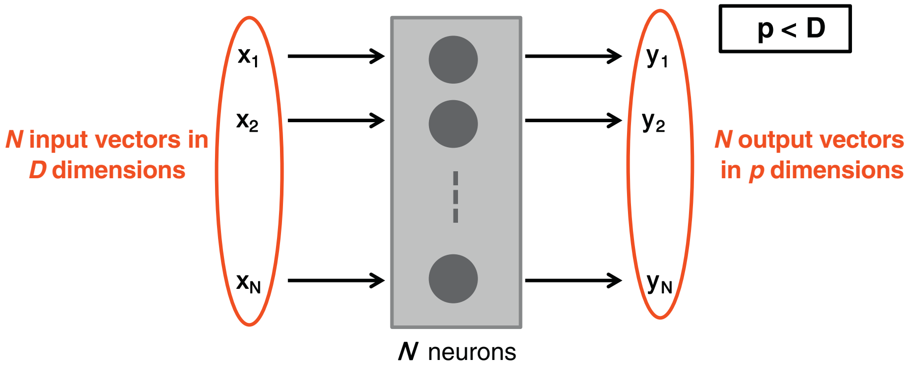

Fig. 1. Curvilinear Component Analysis.

and all y_j points placed in a distance lower than λ around the y_i point are preferentially moved. The minimization of the E_{CCA} criterion is based on a simple gradient descent which gives the adaptation rule (Eq. (3):

$$\Delta y_j = \propto \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left[2F_\lambda\left(Y_{ij}\right) - \left(X_{ij} - Y_{ij}\right) F'_\lambda\left(Y_{ij}\right)\right]\left(y_i - y_j\right) \tag{3}$$

The most frequently used functions are F1 and F2:

$$F1_\lambda\left(Y_{ij}\right) = \begin{cases} 1, & Y_{ij} \le \lambda \\ 0, & Y_{ij} > \lambda \end{cases} \tag{4}$$

$$F2_\lambda\left(Y_{ij}\right) = \frac{1}{1 + e^{\left(Y_{ij} - \lambda\right)}} \tag{5}$$

For the sample applications presented below, the rectangular function F1 (Eq. (4)), will be used in the CCA algorithm [14]. This step function is interesting because it is positive, decreasing and its derivative is null that implies the possibility to minimize E_{CCA} with a modified stochastic gradient descent (Eq. (6)) easier to compute.

$$\Delta y_j = \alpha F\left(Y_{ij}\right) \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left(y_j - y_i\right) \qquad \forall j \neq i \tag{6}$$

In the case F_λ (Y_{ij}) is a step function only the y_j points placed in a lower distance of λ are moved around the y_i point.

CCA aims to conserve the shortest distances, and we will use this technique to visualize clusters defined by two or more very closely-spaced points. Indeed, two closely-spaced points in the initial space in *D* dimensions will conserve their distances in the space in *p* dimensions; if we choose *p* equal to 2, we will get an image of how the initial points were distributed in the experimental design space or the high dimensional database.

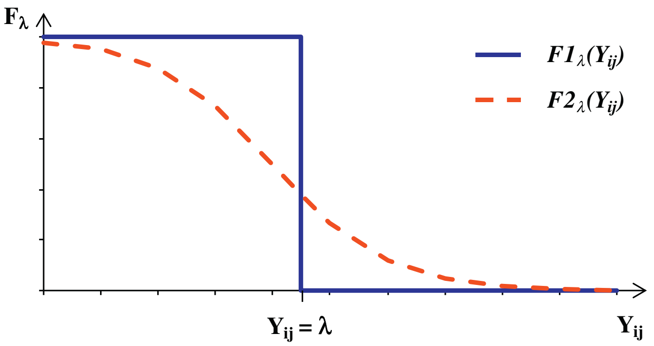

Fig. 2. Examples of the cost function F_λ aiming to favor short distances. Y_{ij} is the output Euclidian distance between two output vectors y_i and y_j, and λ is the neighborhood radius.

2.2.2. Quality criterion

CCA will thus allow visualization of clusters, but it also appears relevant to have a criterion indicating the presence of clusters. For this, we constructed a reference experimental design in 2 dimensions with an identical number of points to the number on the designs to be assessed. This reference design is generated by applying the WSP algorithm to uniformly distribute points throughout a space, whatever its dimensionality. The criterion *R* (Eq. (7)) corresponds to the ratio of the shortest minimal distances between two points (known as *Mindist*) for the CCA design and the reference design.

$$\mathrm{R} = \frac{\text{Mindist for the design after CCA projection}}{\text{Mindist for the reference design}} \tag{7}$$

The value for this ratio is an indicator of the quality of point distribution. If *R* tends towards 1, the design is very close to the reference and the distribution can be considered uniform; in contrast, if *R* tends towards 0, the experimental design contains clusters.

3. Results and discussion

3.1. 2 dimensional example

In the first instance, we used the cluster-detection and gap-filling method on a two-dimensional example to make it possible to track the elimination and/or addition of points visually. To do this, CCA – which was developed for very high dimensions – was not very useful, and we only applied the WSP algorithm. Thus, we developed a random two-dimensional design with 80 points (Fig. 3a.). This design is poorly distributed as it contains both clusters and gaps, and therefore has poor intrinsic properties: a low *Mindist* value (0.006) characteristic of the presence of clusters and a high *Coverage* (0.467) which indicates heterogeneous distances (Table 1). In parallel, we used the WSP algorithm to build a reference uniform experimental design with 80 points, to determine the d_{min} value ($d_{min} = 0.108$) below which two points would be considered too closely-spaced. These points will be eliminated (Fig. 3b.).

Through this process, 47 points were eliminated, leading to an improvement in the uniformity of the design, characterized by an increase in the *Mindist* value (from 0.006 to 0.110) and a reduction in the *Coverage* value (from 0.467 to 0.191). The Box Plots [28] for the minimal distances also show this improvement, by comparing the minimal distances for the reference design with 80 points (Fig. 4a.) and the initial design with clusters and gaps (Fig. 4b.) to the designs after repair (Fig. 4c. and d.).

We know that with a Space Filling Design, the distances between points are homogeneous. This results in a very reduced interquartile range, resulting in superposition of the minimal and maximal values in the Box Plot (Fig. 4a.).

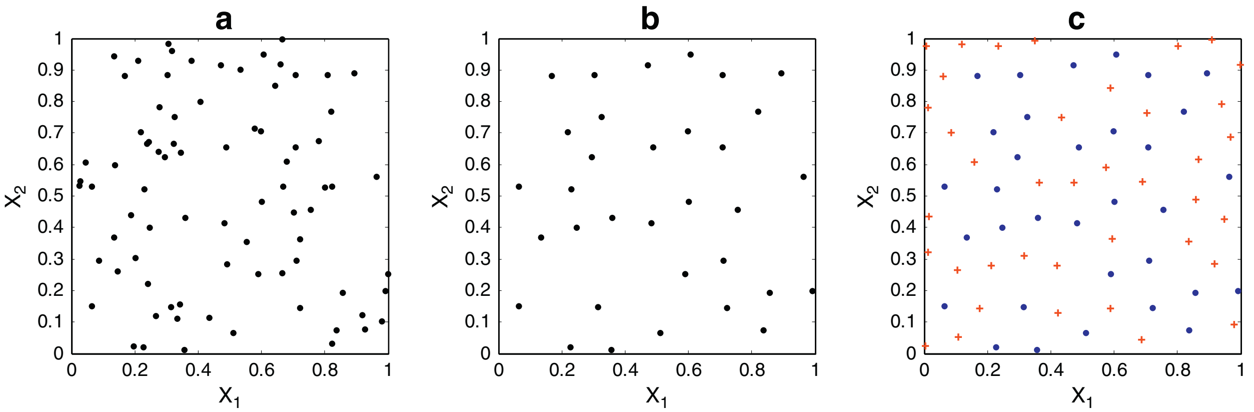

Fig. 3. Graphical representation of point distribution with: (a.) initial design, (b.) design without clusters, and (c.) repaired design.

On the other hand, the Box Plot for the random design (Fig. 4b.) shows a high degree of dispersion of the minimal distances, indicating the presence of clusters and gaps.

The representation of the minimal distances for the random design after elimination of clusters (Fig. 4c.) shows a shift towards greater distances, with a minimal value equal to that of the reference design. However, the total range remains broad, due to the presence of gaps. The next step will therefore be to fill these gaps. To do this, we protected the 33 points remaining after cluster elimination, and added a group of 1000 candidate points. The WSP algorithm was then applied using the same d_{min} value as the reference design. The design repaired by cluster-suppression and gap-filling, contains 72 points and presents good uniformity criteria (*Mindist* = 0.108 and *Coverage* = 0.050), close to those of the reference design (Table 1, Fig. 3d.).

3.2. 20 dimensional example

After studying an example in 2D, we now wish to apply these experimental design repair methods to cases with higher dimensionality. Construction of SFD relies on algorithms which are effective in low dimensions ($D < 10$), but whose quality decreases with increasing dimensionality (20D, 30D, etc.) due to the appearance of empty zones (curse of dimensionality [29]) and/or clusters.

For this study, we constructed several designs in 20D with 200 points: random design and SFD such as low-discrepancy sequences (Sobol' sequence and Faure sequence). Low-discrepancy sequences [8–12] use deterministic algorithms to obtain a uniform distribution of points based on the discrepancy criteria [30] measuring the distance between an empirical distribution of data points and a theoretically uniform distribution of points. Previous studies [18] have shown that the latter are not optimal in terms of uniformity (alignments, lacunae and motifs could appear) as the number of dimensions increases. We also constructed a design with clusters (the points constituting the clusters are represented by red crosses), generated by adding very closely-spaced points to a SFD (see Fig. 5). For each design, the standard uniformity criteria, *Mindist* and *Coverage*, can be calculated (Table 2).

The low *Mindist* values for some designs indicate a very close spacing between points. A simple graphical representation to visualize these distributions is of very limited interest, as it is only possible to view sections of the design (Fig. 5) making it impossible to visualize clusters.

Table 1
Mindist and *Coverage* values for the random experimental 2D design before and after repair.

	Number of points	*Mindist*	*Coverage*
Reference design	80	0.108	0.033
Random design	80	0.006	0.467
Design after cluster suppression	33	0.110	0.191
Repaired design	72	0.108	0.050

If CCA is applied to the 20D space the points can be projected onto a two-dimensional space (Fig. 4). The reduction in dimensionality using CCA, given that short distances are conserved, will allow any clusters present to be rapidly identified.

In Fig. 4a. and b. no clusters are observed but we cannot conclude about the quality of the spread of points. Fig. 4d. clearly shows a regrouping of points corresponding to the voluntarily added clusters, and Fig. 6c. shows the known alignments of points in the Faure series at high dimensions. To complete this visual information, we calculated the *R* ratio: to do this, the *Mindist* for a uniform reference design in 2D with 200 points (*Mindist* = 0.066) was compared to that of each design after CCA projection (Table 3).

This analysis reveals a wide variation in ratio values, indicating that the designs considered are not equivalent in terms of uniformity. The poorest designs are: Faure sequence and the design with clusters, which both present a very low *R* ratio, confirming the presence of the clusters detected in the graphical representation (Fig. 6).

3.2.1. Repairing 20D designs

3.2.1.1. Step 1: Cluster elimination. To repair these designs, we can start by eliminating clusters. In this case, the reference d_{min} value must be derived from a reference 20D design with 200 points. This will be used to detect clusters and to eliminate all the points located at a shorter distance than this reference distance. Results for this step are reported in Table 4, where we observe that the number of points remaining after cluster suppression is very low for low-discrepancy sequences. This indicates that these designs always present accumulations of points in specific zones of the factor space. We also observe that for the design with clusters, this step successfully eliminated the voluntarily created clusters.

3.2.1.2. Step 2: Filling gaps. After eliminating the clusters, the remaining points are considered to be protected points. To these, we add a design with 10,000 candidate points. We then applied the WSP selection algorithm to fill the gaps using these candidate points, based on the procedure described in paragraph 2.1.4. This step results in the addition of many points to designs which had large numbers of clusters, such as low-discrepancy sequences, and random designs to a lesser extent.

3.2.1.3. Step 3: Application of the CCA and ratio calculation. Once the clusters had been eliminated and the gaps filled, the various designs were projected by CCA to calculate the *Mindist* in the 2D projection space. For each design, the *R* ratio of the *Mindist* was calculated taking the reference *Mindist* for an equivalent number of points in 2D (Table 4).

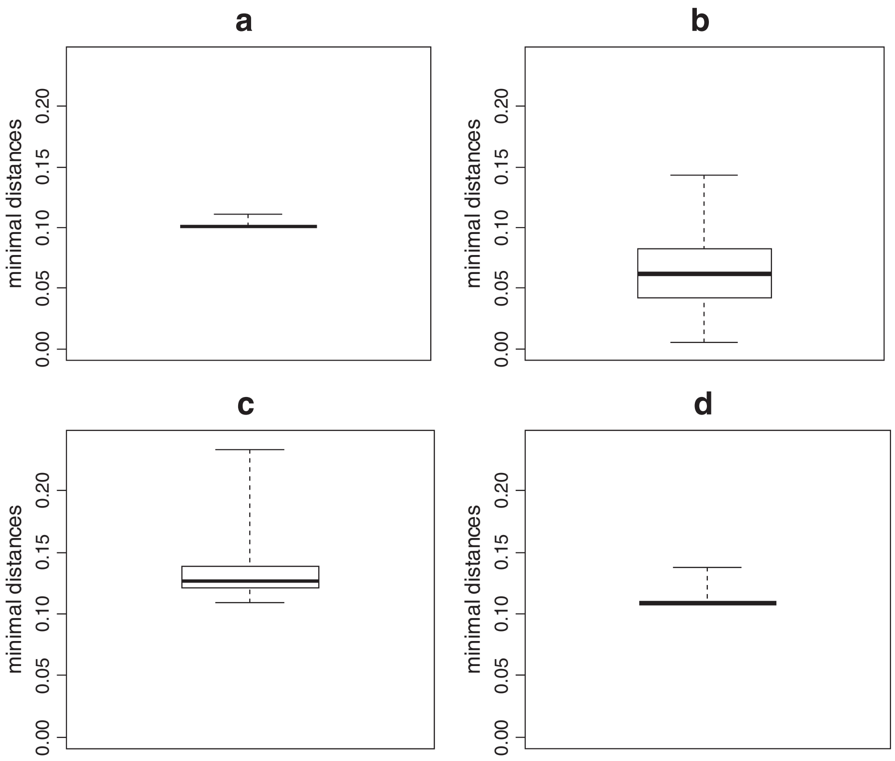

Fig. 4. Box Plot representations of the minimal distances for the 2D design with 80 points. (a.) for a reference design, (b.) for the random design with clusters and gaps, (c.) for the design after elimination of clusters, and (d.) for the fully repaired design.

Through these three successive steps, it was possible to eliminate clusters, fill gaps and finally measure a criterion to assess the quality of the point distribution after repair.

All the repaired designs present ratios close to 0.80 (Table 4), thus guaranteeing a good distribution of the points throughout the factor space. Thus, designs such as the Faure sequence and designs with clusters, which were initially considered bad, can be repaired using this method. However, since very few of the initial points remain after cluster suppression from Sobol' and Faure sequence designs, the ratios after repair should be interpreted with caution as, in these cases, the gap-filling phase could in fact be considered a design-reconstruction step.

From results in Table 4, a comment may be added about the elimination of points. In Fig. 6a. and b, no clusters are observed but many points are eliminated by the WSP algorithm. This phenomenon can be explained by very near points, but not necessary as clusters. For example the points can be only very near and aligned. To detect this phenomenon, the best way is to compare the initial *Mindist* criterion (Table 2: *Mindist* = 0.867 and 0.925 for respectively the random design and Sobol' sequence) to *Mindist* of the reference design in 20D with 200 points which is equal to 1.527. *Mindist* of random design and Sobol' sequence in 20 dimensions are lower than *Mindist* of reference design that means that the spread of points of these designs is not uniform. Thus, the points are closer than the points of reference design, but not appear as clusters.

3.3. Repairing experimental designs after folding into a sub-space

The first step in the study of a complex phenomenon is often a sensitivity analysis. This consists in rapidly identifying the most important variables so as to eliminate those not affecting the response. This step

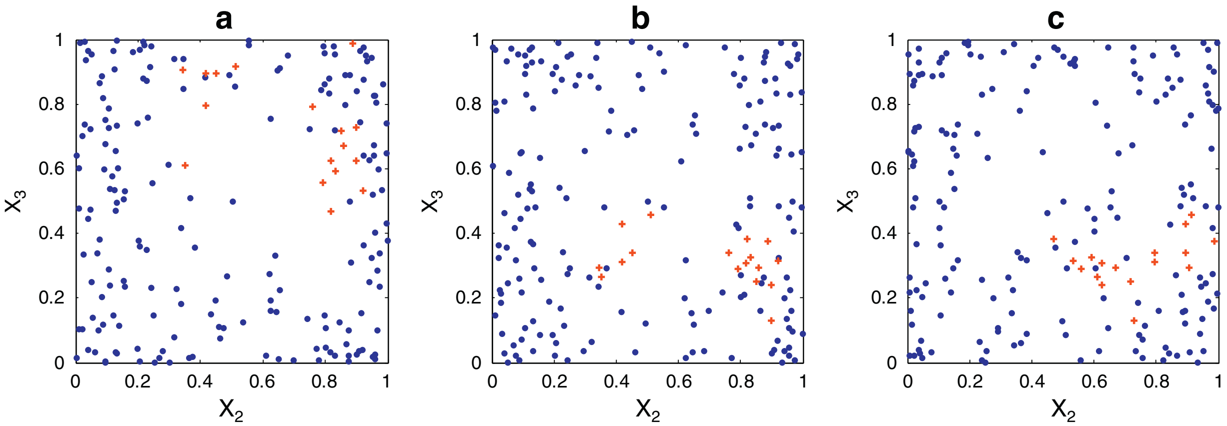

Fig. 5. Design with clusters in 20 dimensions. (a.) viewed in the (X_1, X_2) plane, (b.) in the (X_1, X_3) plane, and (c.) in the (X_2, X_3) plane. The red crosses represent the points constituting the clusters generated by adding very-close points to a WSP design.

Table 2
Mindist and *Coverage* values for the designs studied in 20D with 200 points.

20D—200 points	*Mindist*	*Coverage*
Random design	0.867	0.094
Sobol' sequence	0.925	0.087
Faure sequence	0.194	0.521
Design with clusters	0.315	0.215

Table 3
Mindist criteria and ratios after CCA projection.

	Mindist after CCA	*Mindist* of reference design 2D—200 points	Ratio *R*
Random design	0.026	0.066	**0.40**
Sobol' sequence	0.037		**0.56**
Faure sequence	0.007		**0.10**
Design with clusters	0.013		**0.20**

is crucial when a very large number of potentially influential variables exist, and where only a small number of variables are to be kept for the subsequent steps through work in a sub-space. This step, known as refolding, may modify how the points are distributed across the experimental design. The distribution might be uniform in the initial space, but non-uniform in the sub-space of retained variables. To illustrate this case, we chose to study a function that is widely used in the literature, the g-Sobol' function [31,32], which is defined by the following relationship (Eq. (8)), whatever the k dimensions:

$$y = \prod_{j=1}^{k} g_j\left(x_j\right) \tag{8}$$

with $g_j\left(x_j\right) = \frac{|4x_j-2|+a_j}{1+a_j}$ where $a_j = \{0,1,4,5,9,99,99,99,99\}$ and $x_j \in [0,1]$.

In this case g-Sobol' function is studied in a space with 8 dimensions, thus $k = 8$ in the Eq. (8). The section view of this function (Fig. 7) shows its specificity: pronounced irregularities.

For this study, we constructed a WSP design with 1600 points. We first performed a sensitivity study according to the Improved Sensitivity THrough Morris Extension method called ISTHME [33], which is based on classical Morris's method [34] but uses any set of points and more particularly a uniform design. This method allows the classification of factors in three groups: factors having (1) negligible effects, (2) linear and additive effects or (3) nonlinear or interaction effects. As in classical Morris's method, elementary effects $d_j(y)$ (Eq. (9)) are calculated for each factor X_j:

$$d_j(y) = \frac{y\left(x_1,\dots,x_{j-1},x_j+\Delta_j,x_{j+1},\dots,x_k\right)-y(x)}{\Delta_j} \tag{9}$$

where Δ_j is a value in $\{1/(p-1),\dots,1\text{-}1/(p-1)\}$, with p the number of levels.

According to the values of the mean of the absolute value of elementary effects $\mu^*_j(y)$ and the standard deviation $\sigma_j(y)$, the factors are classified as follows:

- low values of $\mu^*_j(y)$ and $\sigma_j(y)$ characterize factors with negligible effects (1),
- a high value of $\mu^*_j(y)$ and a low value of $\sigma_j(y)$ characterize factors with linear effects (2),
- high values of $\mu^*_j(y)$ and $\sigma_j(y)$ characterize factors with nonlinear or interaction effects (3).

In this study, the ISTHME method shows that only two variables X1 and X2 are detected as influential. The initial design in 8 dimensions with 1600 points was then "refolded" in the (X1, X2) sub-space and

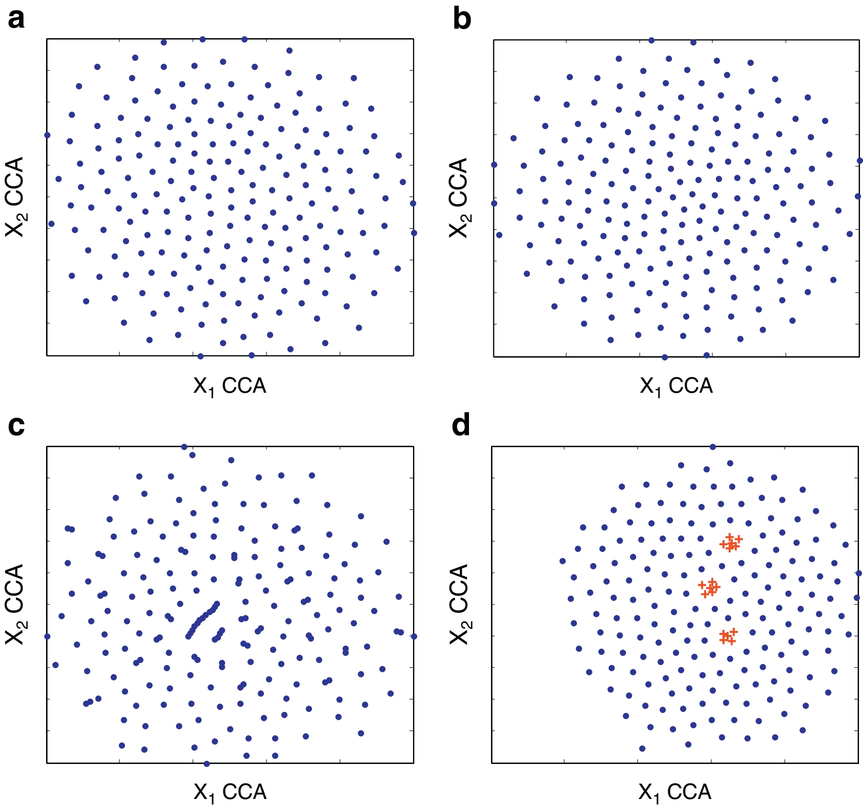

Fig. 6. CCA applied to high dimensionality designs (20D—200 points). (a.) random design, (b.) Sobol' sequence, (c.) Faure sequence, and (d.) design with clusters. All designs are represented in the projection space.

Table 4
Comparing ratios before and after repairing 20D designs.

	Ratio *R* for the initial designs	Number of points		*Mindist* after CCA and full repair	*Mindist* reference for the number of points after full repair	Ratio *R* after full repair
		After cluster elimination	After filling gaps			
Random design	**0.40**	42	188	0.054	0.066	**0.82**
Sobol' sequence	**0.56**	6	200	0.053	0.066	**0.80**
Faure sequence	**0.10**	3	199	0.052	0.064	**0.81**
Design with clusters	**0.20**	185	211	0.053	0.062	**0.85**

must be assessed to determine its uniformity before subsequent use. To do this, a uniform design with 1600 points in 2 dimensions was generated as a reference design. Its d_{min} value will be considered as the ideal minimal distance between two points. Using this to eliminate clusters retains 863 of the 1600 initial points; these remaining points are the protected points. The gaps can then be filled using the WSP algorithm to add 615 points. To illustrate this distribution, we represent the minimal distances as a Box Plot for each step (Fig. 8).

As the design is repaired, the minimal distances improve, and therefore the point distribution throughout the space becomes more uniform. Indeed, the Box Plot is refined, showing a closer correlation between the quantile values, which eventually stabilize at close to the values of the uniform reference design.

4. Conclusion

Space Filling Design is increasingly used, particularly in the field of numeric simulation, but its performance in higher dimensions remains limited. Indeed, studies have shown that algorithms which are powerful at low dimensions ($D < 10$) become difficult to use at higher dimensions ($D > 20$ or 30); this leads to poor uniformity. This non-uniform distribution in the factor space can lead to accumulation of points in specific zones, or to the appearance of gaps, which can penalize studies of the response surface. We show here that Curvilinear Component Analysis can be used to detect the presence of clusters, which can then be eliminated using the WSP algorithm if necessary. WSP can also be used to fill zones where gaps are present. These methods were tested on various cases,

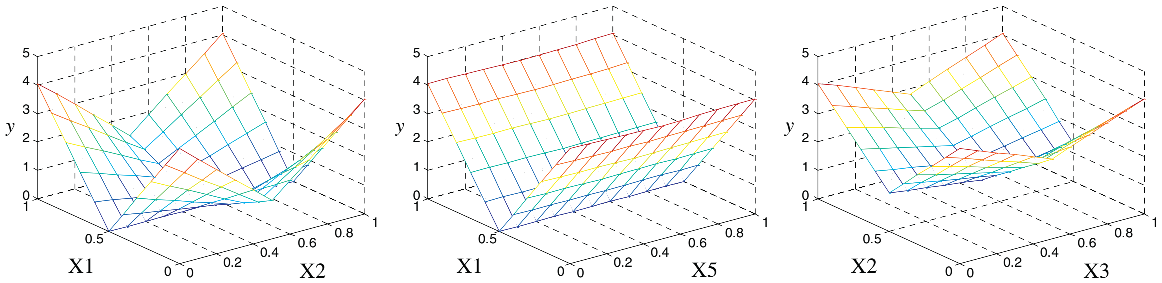

Fig. 7. Section views representative of the output variables obtained using the g-Sobol' function.

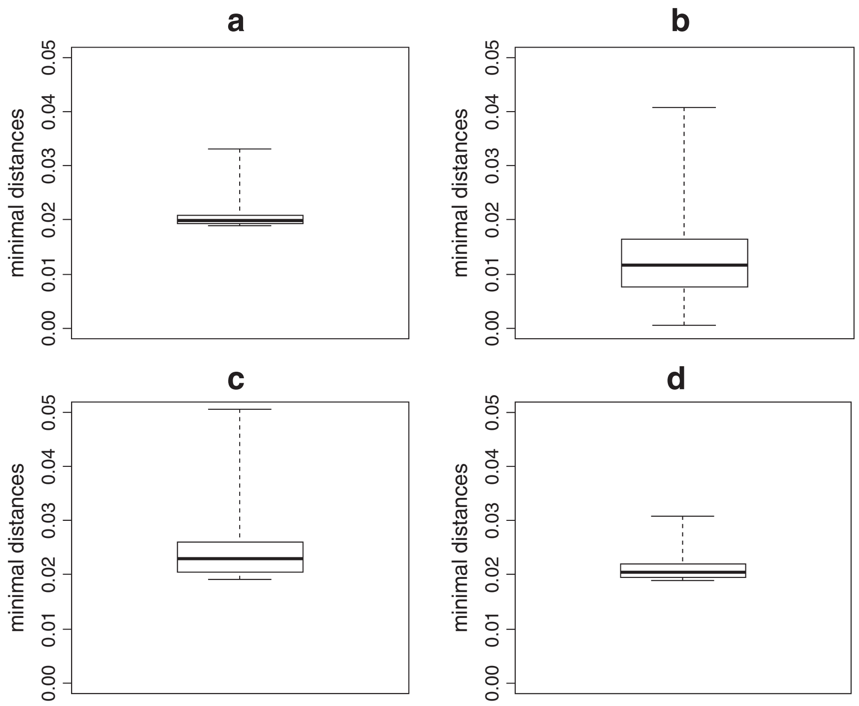

Fig. 8. Box Plot representations of minimal distances. (a.) for the reference design, (b.) for the “refolded” design, (c.) for the “refolded” design after cluster elimination, and (d.) for the “repaired” design.

and are shown to be promising while also remaining easy to use and rapid to implement, even at higher dimensions.

Conflict of interest

None.

References

[1] K.T. Fang, D.K.J. Lin, P. Winker, Y. Zhang, Uniform design: theory and application, Technometrics 42 (2000) 237–248.

[2] K.T. Fang, R. Li, A. Sudjianto, Design and Modeling for Computer Experiments, Chapman & Hall/CRC, 2006.

[3] T.J. Santner, B.J. Williams, W. Notz, The Design and Analysis of Computer Experiments, Springer, 2003.

[4] M.E. Johnson, L.M. Moore, D. Ylvisaker, Minimax and maximin distance designs, J. Stat. Plan. Infer. 26 (1990) 131–148.

[5] V.C.P. Chen, K.-L. Tsui, R.R. Barton, M. Meckesheimer, A review on design, modeling and applications of computer experiments, IIE Trans. 38 (2006) 273–291.

[6] M.W. Trosset, Approximate maximin distance designs, Proc. Sect. Phys. Eng. Sci, 1999, pp. 223–227.

[7] M. Gunzburger, J. Burkhardt, Uniformity measures for point samples in hypercubes, http://people.sc.fsu.edu/jburkardt/pdf/ptmeas.pdf 2004.

[8] H. Faure, Discrépances de suites associées à un système de numération (en dimension un), Bull. Soc. Math. Fr. (1981) 142–182.

[9] J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numer. Math. 2 (1960) 84–90.

[10] J.M. Hammersley, Monte Carlo methods for solving multivariable problems, Ann. N. Y. Acad. Sci. 86 (1960) 844–874.

[11] I.M. Sobol, On the distribution of points in a cube and the approximate evaluation of integrals, USSR Comput. Math. Math. Phys. 7 (1967) 86–112.

[12] I.M. Sobol, Uniformly distributed sequences with an additional uniform property, USSR Comput. Math. Math. Phys. 16 (1976) 236–242.

[13] P. Demartines, Analyse de données par réseaux de neurones auto-organisés, (Thesis) Institut National Polytechnique de Grenoble, 1994.

[14] P. Demartines, J. Herault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets, IEEE Trans. Neural Netw. 8 (1997) 148–154.

[15] M. Sergent, Contribution de la Méthodologie de la Recherche Expérimentale à l'élaboration de matrices uniformes: application aux effets de solvants et de substituants, (Thesis) 1989. (Marseille).

[16] M. Sergent, R. Phan Tan Luu, J. Elguero, Statistical analysis of solvent scales, part 1, An. Quím. Int. Ed. 93 (1997) 71–75.

[17] M. Sergent, R. Phan Tan Luu, R. Faure, J. Elguero, Statistical analysis of solvents scales, part 2, An. Quím. Int. Ed. 93 (1997) 295–300.

[18] J. Santiago, M. Claeys-Bruno, M. Sergent, Construction of space-filling designs using WSP algorithm for high dimensional spaces, Chemom. Intell. Lab. Syst. 113 (2012) 26–31.

[19] J. Santiago, Développement de nouveaux plans d'expériences uniformes adaptés à la simulation numérique en grande dimension, (Thesis) Aix Marseille, 2013.

[20] A. Beal, M. Claeys-Bruno, M. Sergent, Constructing space-filling designs using an adaptive WSP algorithm for spaces with constraints, Chemom. Intell. Lab. Syst. 133 (2014) 84–91.

[21] N.A. Butler, Optimal and orthogonal Latin hypercube designs for computer experiments, Biometrika 88 (2001) 847–857.

[22] A.B. Owen, Orthogonal arrays for computer experiments, integration and visualisation, Stat. Sin. 2 (1992) 439–452.

[23] M. Stein, Large sample properties of simulations using Latin hypercube sampling, Technometrics 29 (1987) 143–151.

[24] B. Tang, Orthogonal array-based Latin hypercubes, J. Am. Stat. Assoc. 88 (1993) 1392–1397.

[25] B. Tang, A theorem for selecting oa-based latin hypercubes using a distance criterion, Commun. Stat.-Theory Meth. 23 (1994) 2047–2058.

[26] K.Q. Ye, Orthogonal column Latin hypercubes and their application in computer experiments, J. Am. Stat. Assoc. 93 (1998) 1430–1439.

[27] J. Franco, Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes, (Thesis) Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008.

[28] J.W. Tukey, Exploratory data analysis, 1977.

[29] M. Köppen, The curse of dimensionality, 5th Online World Conf. Soft Comput. Ind. Appl, 2000, pp. 4–8.

[30] E. Thiémard, Sur le calcul et la majoration de la discrépance à l'origine, (Thesis) Ecole Polytechnique fédérale de Lausanne, 2000.

[31] A. Saltelli, I.M. Sobol', About the use of rank transformation in sensitivity analysis of model output, Reliab. Eng. Syst. Saf. 50 (1995) 225–239.

[32] A. Saltelli, E.M. Scott, Sensitivity Analysis, J. Wiley & Sons, New York; Chichester; Weinheim, 2000.

[33] J. Santiago, B. Corre, M. Claeys-Bruno, M. Sergent, Improved sensitivity through Morris extension, Chemom. Intell. Lab. Syst. 113 (2012) 52–57.

[34] M.D. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (1991) 161.