Chemometrics and Intelligent Laboratory Systems 133 (2014) 84–91

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Constructing space-filling designs using an adaptive WSP algorithm for spaces with constraints☆

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

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

ARTICLE INFO

Article history:
Received 26 August 2013
Received in revised form 7 November 2013
Accepted 24 November 2013
Available online 1 December 2013

Keywords:
Experimental design
Space-filling designs
High dimensional space
WSP algorithm

ABSTRACT

Not all phenomena can be studied using standard experimental designs. Indeed, non-linear phenomena require experimental designs to cover the whole variable space in a reasonable number of experiments. Space-filling designs (SFD) propose a uniform distribution of points and are well adapted to numerical simulations. However, not all SFDs are equivalent in terms of uniformity of point distribution throughout the variable space, as assessed by quality criteria (such as MinDist, Coverage, etc.) and many algorithms which are powerful in low dimensional spaces ($D < 10$) become difficult to use at higher dimensions (20D, 30D, etc.). The Wootton, Sergent, Phan-Tan-Luu's algorithm (WSP) was developed to select points from a set of candidate points and generate designs with good uniformity criteria whatever the number of dimensions. This study presents an adjustment of this algorithm, called adaptive WSP to obtain designs with specific experimental constraints, or when density is to be increased in a zone of particular interest. This adaptive WSP algorithm will be very useful as the number of dimensions increases and can solve the problem of the "hollow" center.

1. Introduction

In many fields, computer simulations are used to replace often costly laboratory experiments. These models simulate complex phenomena and are increasingly realistic. Thus, despite the growing calculation capacity of processors, calculation times remain long. These models present an increasingly realistic picture, which can be as difficult to interpret as real-life experiments due to the numerous parameters involved, for which the effects may be complex. When complex phenomena are studied, the most commonly used experimental designs may no longer be effective. It is therefore necessary to develop specific designs to study broad spaces with a reasonable number of experiments. Uniform designs (space-filling designs, SFD) [1–3] propose a uniform distribution of points throughout the variable space. These are appropriate for numerical simulations [4–8]. However, not all SFDs are equivalent in terms of quality criteria measuring the uniformity of point distribution, such as the MinDist values [9–11], coverage [12], discrepancy [13], etc. Many algorithms which are powerful in low dimensional spaces ($D < 10$) become difficult to use at higher dimensions (20D, 30D, etc.). Difficulties arise both from the number of input variables, which can be very large, and from the complexity of the phenomena to be modeled in a very broad space. Thus, low discrepancy sequences, such as Faure sequences [14] present very poor uniformity criteria over larger dimensions, with low MinDist values and high coverage. This results in the appearance of clusters, lacunae and/or alignments. In addition, some designs, such as Strauss designs [15] can present good uniformity criteria, but are time-consuming to build as the number of dimensions increases. Consequently, current uniform designs do not meet the requirements for building uniform designs of experiments with high dimensionality.

We recently proposed a new construction algorithm, the Wootton, Sergent, Phan-Tan-Luu's algorithm [16–18] called WSP which can be used to develop designs presenting good uniformity criteria, whatever the number of dimensions and points [19]. In this study, we propose an adjustment of this algorithm, adaptive WSP (aWSP), which will allow specific spaces to be considered. In particular, this adaptation will allow:

- experimental designs to be constructed for all spaces, when the zone of interest is restricted to part of a space or when experimental constraints forbid the study of a particular zone (feasibility problem),
- an increase in density in certain zones of particular interest and/or in which the phenomenon may present non-linearities.

2. Space-filling designs

2.1. Standard SFD

The most frequently used computer designs are *space-filling designs*; these differ from standard experimental designs due to the position of the points, which are no longer simply located at the extremities of the zones of variation of the input variables, but are distributed as uniformly as possible throughout the experimental space. Several families of SFD exist, which differ by the criterion on which the construction method relies.

☆ 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).

0169-7439/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.chemolab.2013.11.009

Of the families of SFD, Latin hypercube sampling (LHS) [20,21] where specific constraints determine the positions of points in the design for the lower dimensions, in general on the factorial axes or faces is extensively used in simulations. In a LHS of dimension D and with N points, each variable will take N values uniformly distributed over the factorial axes. Nevertheless, despite the most uniform projection possible for the variables on the axes, this design criterion does not guarantee good space filling. Low discrepancy sequences use deterministic algorithms [22,23] to obtain a uniform distribution of points based on the discrepancy criteria measuring the distance between an empirical distribution of data points and a theoretically uniform distribution of points. The low discrepancy sequences include Halton sequences [24], Hammersley sequences [25], Sobol sequences [26,27] and Faure sequences [14]. Previous studies [19] have shown that the latter are not optimal in terms of uniformity as the number of dimensions increases: for example Fig. 1.a,b,c shows that alignments, lacunae and motifs appear in the transverse planes for the X_8 and X_9 variables. A possible alternative, Strauss Designs [15,28], are generated according to Gibbs' point process (Strauss 1975 [29]), which is based on the phenomenon of repulsion between particles. This SFD algorithm assimilates the N points of the design with N particles of the same weight, resulting in a uniform distribution of the points throughout the space, whatever the number of dimensions (Fig. 1d). However, optimization of the design parameters remains difficult and time-consuming, and becomes impossible with very high dimensions. More recently, other algorithms have been described for SFD, such as the WSP algorithm [16–18] which involves selection of points from a set of candidate points. This algorithm allows SFDs to be constructed rapidly, even at very high dimensions ($D > 50$). However, precise studies have shown that these designs suffer from the major problem with high dimensions [30], i.e., an "empty" central zone (Fig. 1e). It has been shown that in a distribution of points over a large number of dimensions, the probability that points are located in the corners of the hypercube tends to 1 as the number of dimensions increases. This can be explained by calculating the ratio of volumes between a hypercube of dimension D and a hypersphere contained within this hypercube.

The volume of a hypercube of side $2r$ and dimension D is given by:

$$C_D = (2r)^D.$$

The volume of a hypersphere of radius r contained within this cube is:

$$S_D = \frac{2^{(D+1)/2}\pi^{D/2}}{D!!} r^D$$

where $\lceil D/2 \rceil$ denotes the largest integer smaller than or equal to $D/2$ and

$$D!! = \begin{cases} \prod_{i=1}^{k} (2i-1), & \text{if } D = 2k-1 \\ \prod_{i=1}^{k} (2i), & \text{if } D = 2k \end{cases}.$$

We show that the ratio of volumes (S_D/C_D) (Fig. 2) tends toward 0 when the dimension tends toward $+\infty$, thus the probability of a point being located within the hypersphere tends toward 0.

This phenomenon has been confirmed for SFD by studying the distribution of projections on the factorial axes. For example, for a WSP design with 205 points in 10D, a graphical representation of the numbers of points projected on each factorial axis shows an excess of points at the extreme intervals and an "emptier" center (Fig. 3).

These observations led us to adapt the algorithm so as to increase the number of points in the center of the space, or more generally, in zones of interest.

2.2. Adaptive WSP

In the WSP selection algorithm, points are selected from a set of candidate points so as to be at a fixed minimal distance (d_{min}) from each point in the defined multidimensional parameters space included in the design.

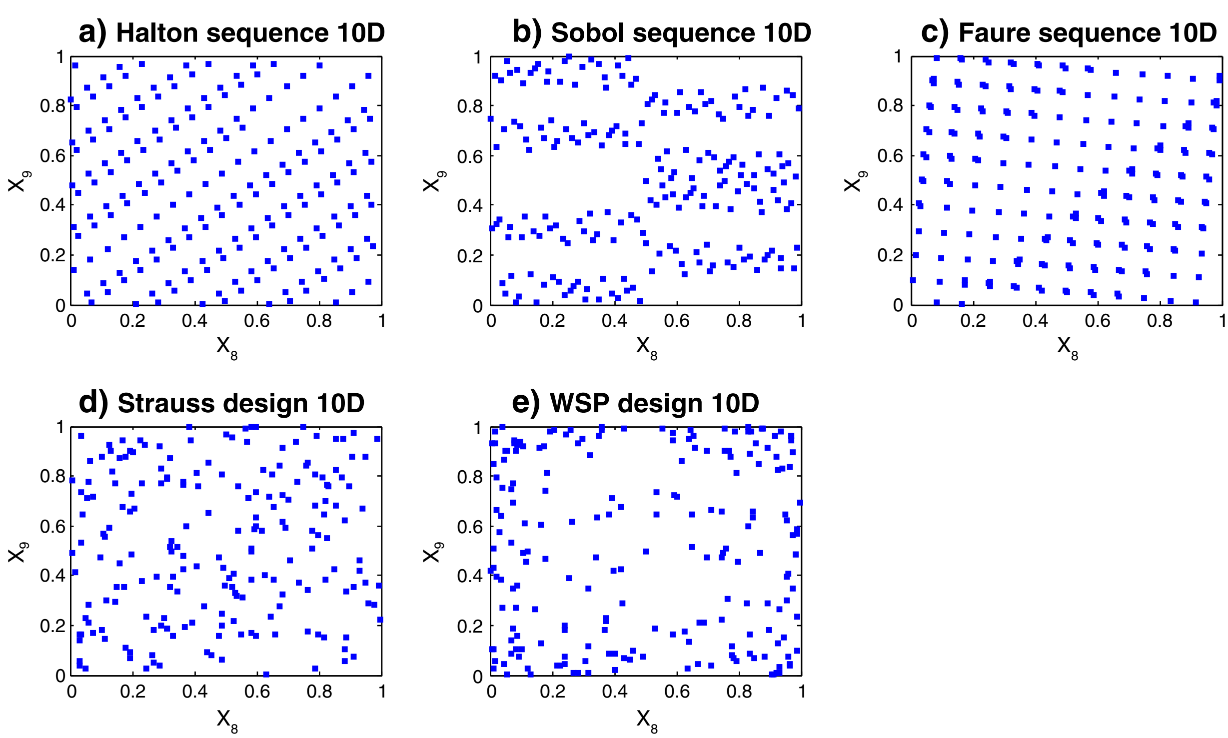

Fig. 1. Projection of points on factorial axes (X_8, X_9) for space-filling designs in 10 dimensions and 200 points: a) Halton sequence, b) Sobol sequence, c) Faure sequence, d) Strauss design, and e) WSP design.

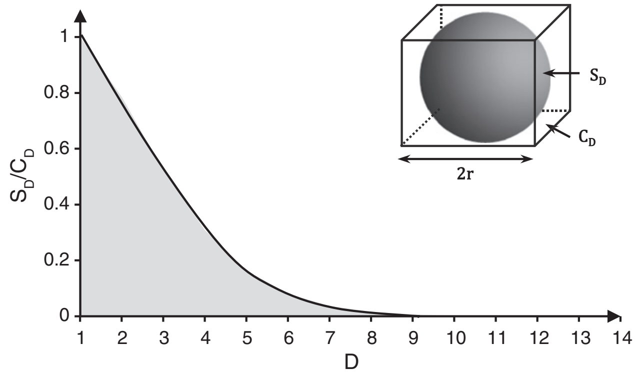

Fig. 2. Ratio of the volumes of unit hypersphere (S_D) and surrounding hypercube (C_D) for dimensions 1 to 14.

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 $D_{OI} < d_{min}$. Point O is eliminated from the set of candidate points and will belong to the final subset.
Step 5 replace point O by the nearest point among the remaining points
Step 6 repeat steps 4 and 5 until there are no more points to choose from.

The number of points in the final subset depends on the value of d_{min} and for a variable density of points within the space, this value will not be constant in the algorithm. Attempts to vary the density of points will require adjustments to this minimal distance (d_{min}) between two points as a function of the position of the point considered within the space.

2.2.1. Adaptive algorithm to increase density at the center of the space

To resolve the problems with construction of space filling designs in high dimensions and to avoid the production of "hollow" centers, we have adapted the WSP algorithm to progressively vary d_{min} to increase the density of points at the center of the space by attributing a lower d_{min} value than that applied in the periphery. In this case, d_{min} will not be constant but will depend on the position of the point considered relative to the center of the space. We suggest using a non-linear relationship to calculate the d_{min} value for each iteration as a function of the position of the point considered:

$$d_{min} = \left[\frac{\text{distance between the point studied and the center} - \text{minimal distance}}{\text{maximal distance} - \text{minimal distance}}\right]^r$$

with, $minimal\ distance = \min_{x_i \in X} dist(x_i, x_{center})$ and $maximal\ distance = \max_{x_i \in X} dist(x_i, x_{center})$ where, $X = \{x_1, x_2, \ldots, x_N\} \subset [0,1]^D$ for a set of N points in D dimensions.

When we wish to increase the density at the center of the space, two d_{min} values must be set. The first, d_{min} *minimum*, will be applied to the center of the space; the second, d_{min} *maximum*, will be the d_{min} value applied to points located in the periphery. These values depend on the whished density of points. The value of the exponent r determines the shape of the curve for variations in d_{min} (Fig. 4). A higher density will be generated in the center compared to the periphery of the space for all values of r. When r is equal to 1, d_{min} will follow a linear variation. For a value of r greater than 1, the variation in d_{min} will favor the selection of points located at short distances, i.e., d_{min} will increase slowly, allowing more points to be conserved close to the center of the space. Conversely, with r less than 1, a very slight increase in density at the center of the space will be generated, and fewer points will be selected in the periphery as longer d_{min} values will be applied for the points located at greater distances from the center, i.e., in the periphery.

Thus, at step 5 of the algorithm presented above, the d_{min} value is recalculated for each iteration as a function of how far the point considered is from the center of the space.

2.2.2. Increasing density in a zone of interest

In some studies, prior knowledge allows a zone of greater interest to be defined. The WSP algorithm can also be adapted to enrich this particular zone of the space. Thus, a first zone can be associated with a short

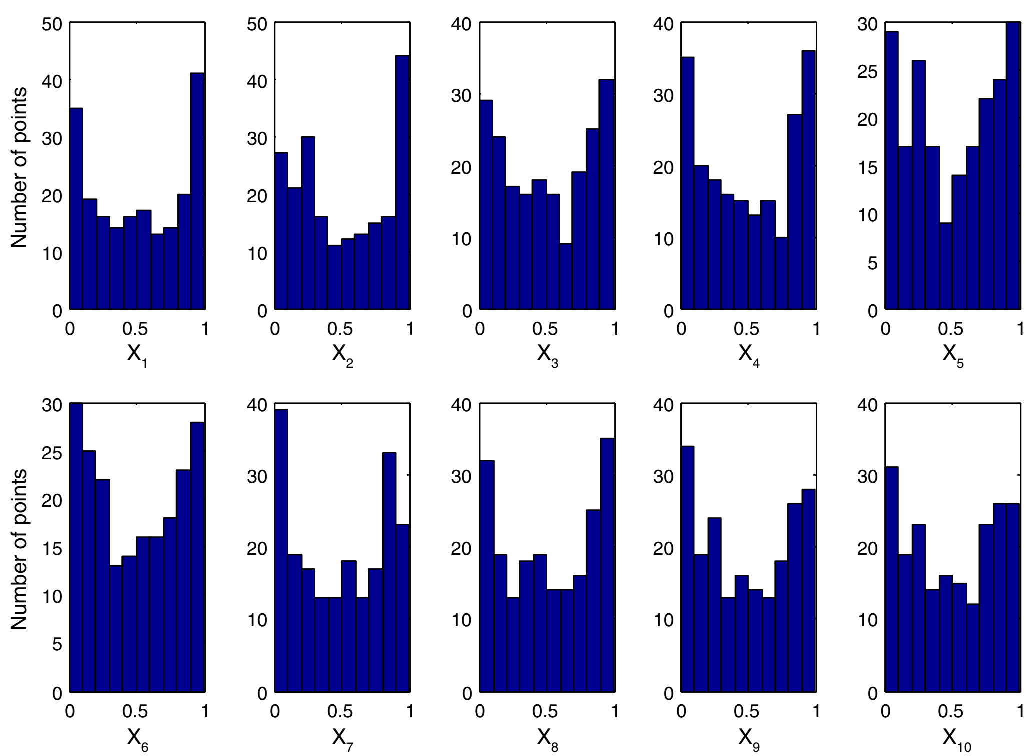

Fig. 3. Histograms representing the number of points on the factorial axes for a design developed in 10D and 205 points using the WSP algorithm.

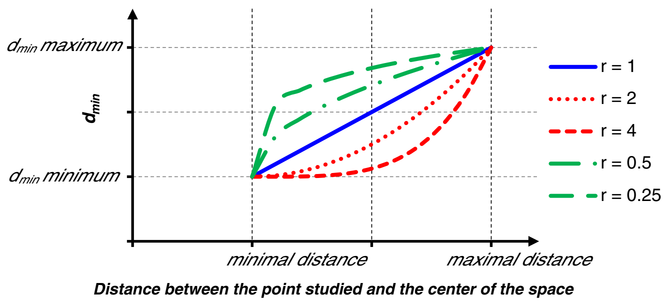

Fig. 4. Examples of the d_{min} function that can be applied to the WSP algorithm to increase the density of points at the center of the space (r indicates the value of the curve for the function).

in-zone d_{min} value, while the remainder of the space is associated with a longer *out-of-zone* d_{min} value.

3. Results and discussion

3.1. Increasing density at the center of the space

3.1.1. Example in 2D

In this example, the candidate design is a random design in two dimensions with 100,000 points. We will compare the matrices obtained by the standard WSP algorithm with $d_{min} = 0.1$ (Fig. 5) and by the adaptive algorithm, aWSP, for which we set d_{min} *minimum* $= 0.04$ and d_{min} *maximum* $= 0.15$ (Fig. 5). The influence of the choice of curvature can be observed on the number of points selected and their distribution throughout the space.

The difference between the two algorithms is shown in Fig. 5. This 2-dimensional example is simply presented as an illustration, since at low dimensions the problem of a hollow center is not encountered. Thus, increasing the density becomes more advantageous in spaces with higher numbers of dimensions. The results show that the standard algorithm distributes the points uniformly throughout the space, while the adaptive algorithm increases the density of points at the center of the space based on the user-defined r coefficient. For $r = 1$, a linear increase in d_{min} is observed from the center of the space toward the periphery. For $r > 1$, the space is populated by applying short d_{min} values for the points located at short and medium distances from the center, resulting in a very high density at the center. For $r < 1$, the increase in density at the center of the space is minimal since the function of variation of d_{min} requires a large d_{min} value to be used for the points located at medium and large distances from the center of the space.

3.1.2. Example in 10D

To demonstrate how our adaptive algorithm performs at higher dimensions, we applied it to a Sobol sequence in 10D with 10,000 points; for this experiment d_{min} *minimum* $= 0.15$, d_{min} *maximum* $= 1.8$, and $r = 1.9$. The final design presents 205 selected points, with a greater density at the center, as shown in the histograms for the number of points (Fig. 6).

Comparing the histograms from Figs. 3 and 6 reveals that the algorithm to increase point density at the center of the space drastically modifies how the points are distributed across the variables' space. Indeed, with the adaptive algorithm the number of points at the center is higher and the number of extreme points (in the periphery of the space) is strongly reduced compared to the standard algorithm.

3.1.3. Example in 20D

When the number of dimensions increases, the problem of "hollow center" is intensified. To illustrate the added value of the adaptive algorithm, an application in 20D with a Sobol sequence and 20,000 points as a candidate set is presented. The graphical representation of the repartition of points on factorial axes allows the comparison of the two algorithms (Fig. 7). The number of histograms being important, only some of them are presented. These graphs highlight the performance of aWSP algorithm that increases the number of points at the center of the domain.

Thus, with this adaptive algorithm we can overcome the curse of dimensionality by choosing to increase the density of points at the center of spaces in high dimensions and this, for any dimension.

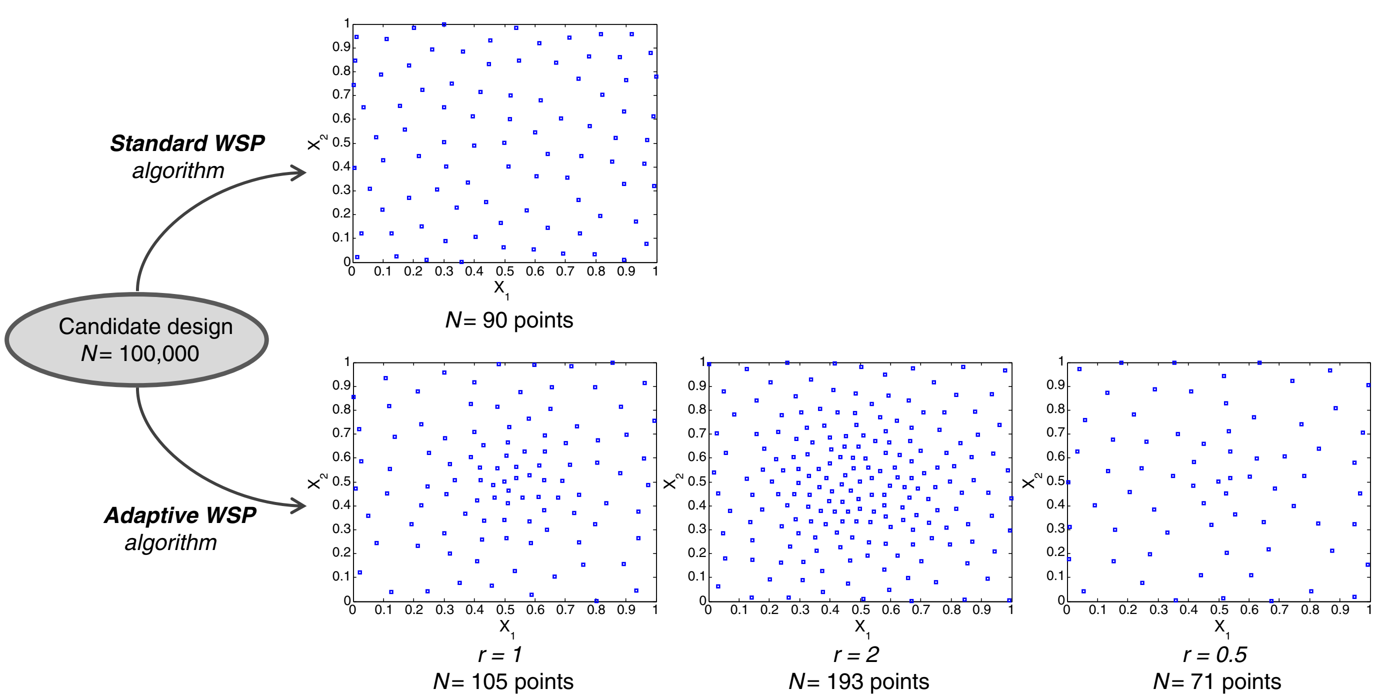

Fig. 5. Comparing standard and adaptive WSP algorithms (with fixed or variable r values).

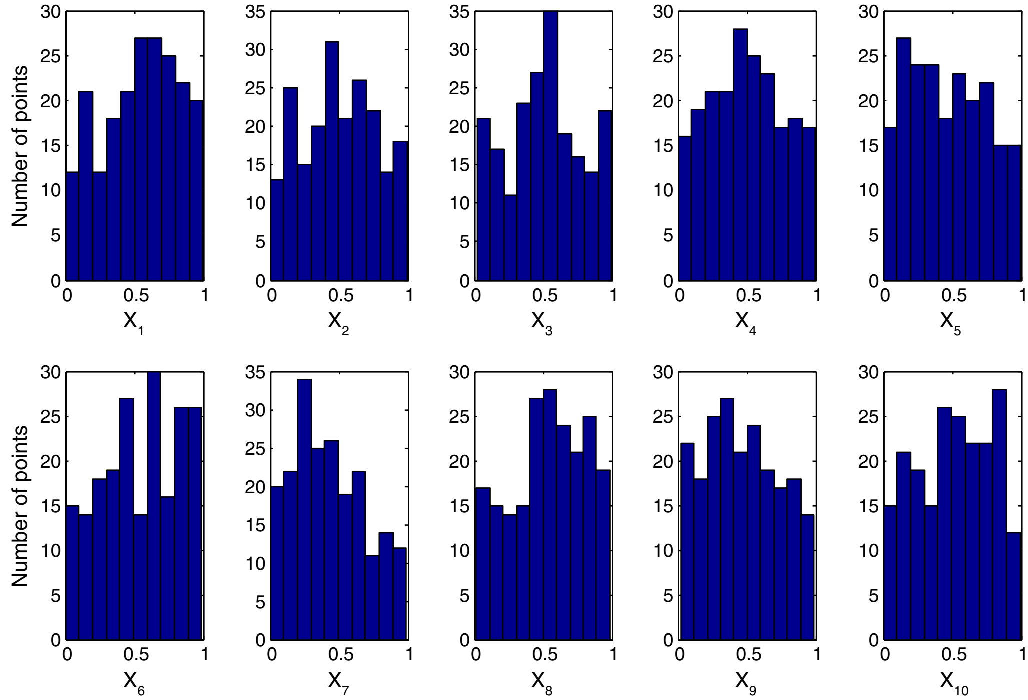

Fig. 6. Histograms representing the number of points on the factorial axes for a design developed using the aWSP algorithm with densification of the center in 10D and 205 points.

3.2. Increasing density in a zone of interest

Sometimes standard space-filling designs cannot be used, particularly due to various constraints (related to time, experiment feasibility, etc.). In these cases, a specific area may need to be defined based on knowledge of the phenomena encountered. Therefore, particular designs must be developed taking these specific constraints into account. Several examples of non-standard designs using adaptive WSP will now be presented and discussed in detail.

3.2.1. Spaces with elementary constraints

For this example, we consider random candidate points in 3D and 100,000 points. The zone of interest, where density is to be increased, is defined by the following constraints: $X_1 \geq 0.6$, $X_2 \geq 0.7$, $X_3 \leq 0.2$. The d_{min} value applied in the zone of interest is 0.04, while in the remainder of the space it is 0.15. This leads to the selection of 524 points (Fig. 8) with a higher density in a specific part of the space.

We next compared the histograms for the numbers of points on each factorial axis for a design produced by a standard WSP, with $d_{min} = 0.15$

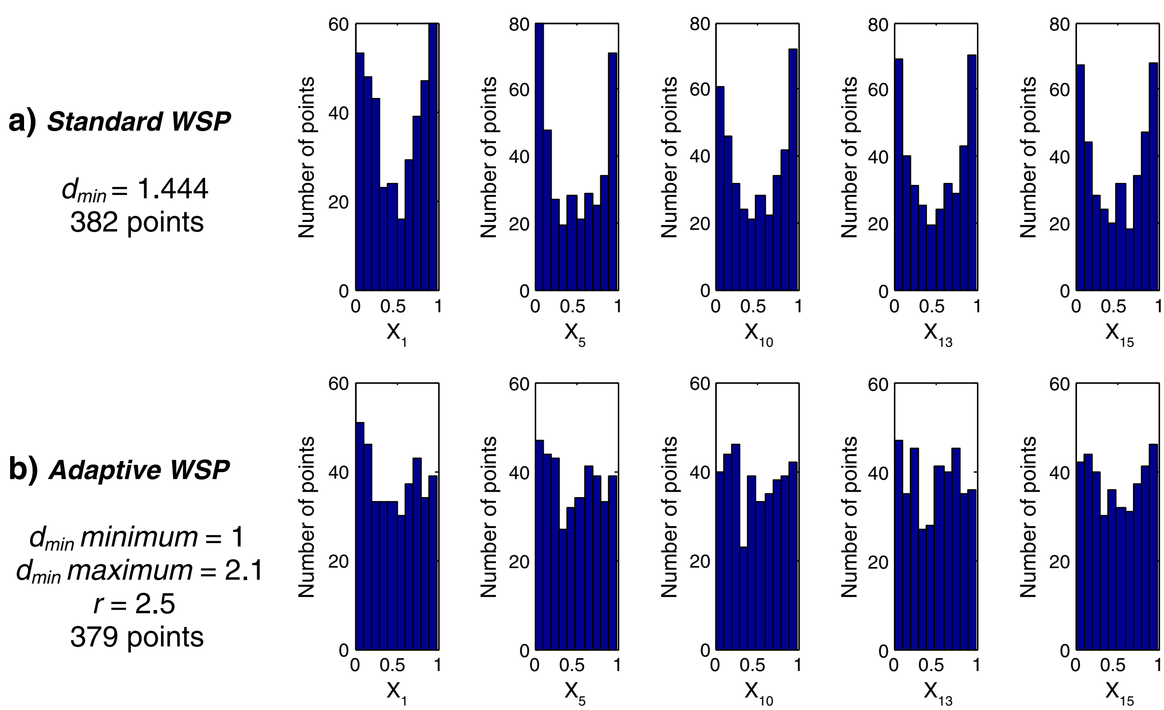

Fig. 7. Histograms representing the number of points on the factorial axes for designs in 20D produced by the standard WSP (a) and by the aWSP algorithm with densification of the center (b). Both methods were applied to a Sobol sequence in 20D with 20,000 points.

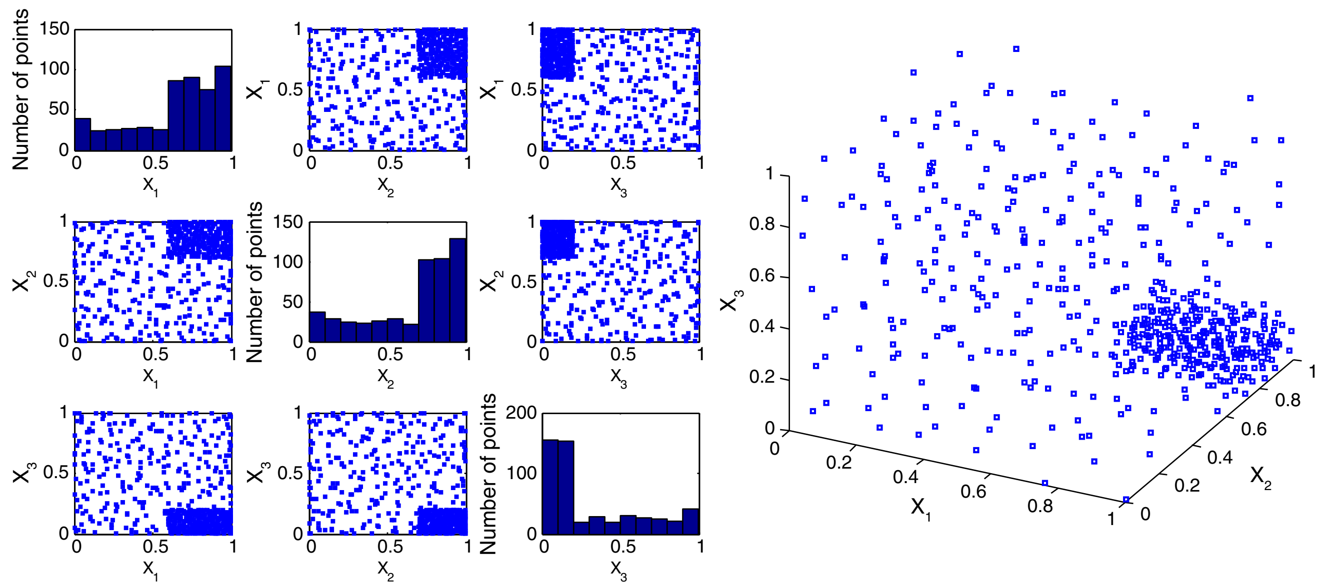

Fig. 8. Increasing density in a zone of interest in an experimental space with three variables. On the left, the different transverse planes are represented. The diagonal shows the histograms representing the number of points for each factorial axis. On the right, the 524 points selected in the 3D space.

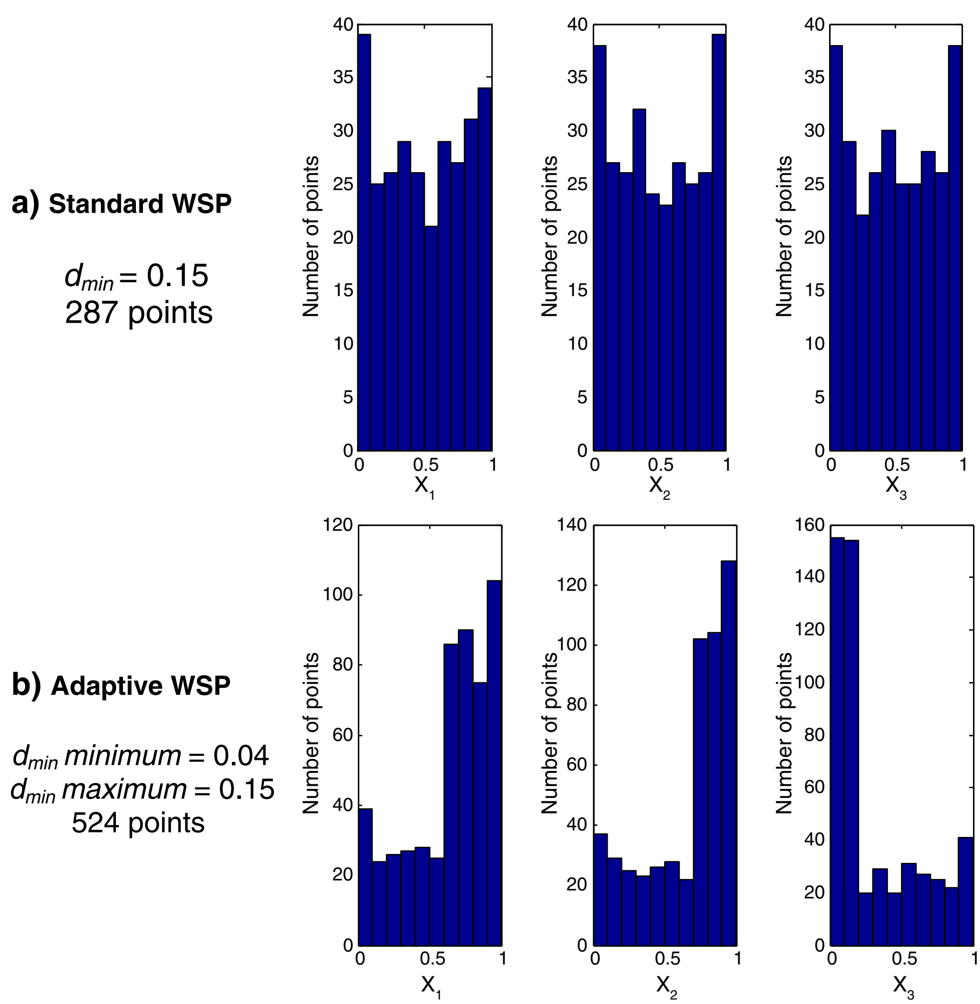

Fig. 9. Histograms representing the numbers of points on the factorial axes for designs produced by a standard WSP (a) and a WSP with increased density in a specific zone of the space (b). Both methods were applied to a random design in 3D with 100,000 points.

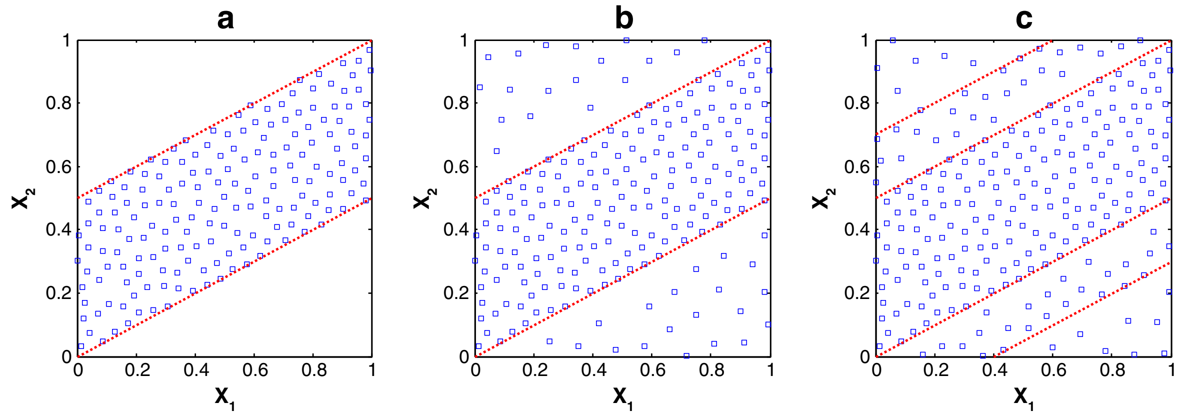

Fig. 10. a) Application of the WSP algorithm to the zone with constraints: $0.5X_1 < X_2 < 0.5X_1 + 0.5$, b) application of the aWSP algorithm with a lower d_{min} value in the zone of interest allows selection of 214 points, c) increasing density in a zone of interest with progressive evolution of the d_{min} value leads to selection of 240 points.

(Fig. 9a) to those for a design produced by the aWSP increasing the density of points in the zone of interest (Fig. 9b). Even though the number of points is not the same in both cases, it is clear that the zone of interest is more densely populated, as shown on the histograms by the higher numbers.

3.2.2. Spaces with relational constraints: non-standard spaces

The constraints set here are relational constraints such as those encountered in real-life cases where the ranges of variation for the input variables are linked. The examples presented below are in two dimensions, but can be generalized to higher dimensions.

When a zone of interest which is restricted to part of a space is to be studied, the constraints for the zone of interest must be defined. For this, we propose use of a standard WSP on a random design in 2D with 100,000 points. We chose to define a zone of interest by the following constraints: $0.5X_1 < X_2 < 0.5X_1 + 0.5$ in which we set $d_{min} = 0.05$ to select 176 points (Fig. 10a).

In the next example, we wished to study the whole space while increasing the density in a zone of particular interest. This contrasts with the previous example where only this zone was populated. Here, we applied an evolutive algorithm, i.e., the d_{min} value is variable across the space. Thus, considering the same candidate design with a d_{min} value for the zone of interest set at 0.05, and a d_{min} *maximum* of 0.1 for the remainder of the space we can select a total of 214 points (Fig. 10b).

We also apply a progressive variation of d_{min}. This results in no change to the zone where the density is to be increased, while an intermediate d_{min} value is applied close to the zone of interest (Fig. 10c). Thus, three zones are defined, each with a distinct d_{min} value: in the zone of interest, $d_{min} = 0.05$; for the part of the space surrounding this zone $d_{min} = 0.07$; and in the remainder of the space $d_{min} = 0.1$. This leads to the selection of 240 points.

4. Conclusion

Space-filling designs are now recognized as suitable for computer experiments, but the main obstacle is with high dimensional spaces ($D > 20$). The calculations used in standard designs take too long or provide a non-uniform spread of points. Therefore, we propose a new type of space-filling design based on the WSP algorithm, the adaptive WSP algorithm ("aWSP"). This can be used for non-standard spaces and to increase the density in a defined part of the space. This new algorithm is very useful as the number of dimensions increases, and can solve the problem of the "hollow" center or take prior knowledge of particular phenomena into account so as to focus on a particular area.

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] O. Vasseur, M. Claeys-Bruno, M. Cathelinaud, M. Sergent, High-dimensional sensitivity analysis of complex optronic systems by experimental design: applications to the case of the design and the robustness of optical coatings, Chin. Opt. Lett. 8 (2010) 21–24.
[5] O. Vasseur, M. Claeys-Bruno, M. Sergent, Quality of high dimensional numerical "Space filling designs" for the sensitivity analysis of interference optical filters, Procedia Soc. Behav. Sci. 2 (2010) 7761–7762.
[6] O. Vasseur, M. Claeys-Bruno, M. Sergent, M. Cathelinaud, Sensitivity analysis of optical coatings manufacturing by numerical experimental designs, Optical Interference Coatings, Optical Society of America, 2010, p. TuA7.
[7] A. Azarian, O. Vasseur, B. Bennaï, V. Jolivet, M. Claeys-Bruno, M. Sergent, et al., Sensitivity analysis of interference optical systems by numerical designs, J. Soc. Fr. Stat. 152 (2011) 118–130.
[8] O. Vasseur, M. Cathelinaud, M. Claeys-Bruno, M. Sergent, Global sensitivity analysis of bandpass and antireflection coating manufacturing by numerical space filling designs, Appl. Opt. 50 (2011) 117–123.
[9] M.E. Johnson, L.M. Moore, D. Ylvisaker, Minimax and maximin distance designs, J. Stat. Plann. Infer. 26 (1990) 131–148.
[10] 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.
[11] M.W. Trosset, Approximate maximin distance designs, Proceedings of the Section on Physical and Engineering Sciences, 1999. 223–227.
[12] M. Gunzburger, J. Burkhardt, Uniformity Measures for Point Samples in Hypercubes, http://people.sc.fsu.edu/~jburkardt/pdf/ptmeas.pdf. (2004).
[13] H. Niederreiter, Low-discrepancy and low-dispersion sequences, J. Number Theory 30 (1988) 51–70.
[14] H. Faure, Discrépances de suites associées à un système de numération (en dimension un), Bull. Soc. Math. Fr. (1981) 142–182.
[15] 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.
[16] M. Sergent, Contribution de la Méthodologie de la Recherche Expérimentale à l'élaboration de matrices uniformes: Application aux effets de solvants et de substituants, 1989. (Thesis, Marseille).
[17] M. Sergent, R. Phan Tan Luu, J. Elguero, Statistical analysis of solvent scales, part 1, An. Quím. Int. Ed. 93 (1997) 71–75.
[18] 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.
[19] 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.
[20] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239.
[21] M. Stein, Large sample properties of simulations using Latin hypercube sampling, Technometrics 29 (1987) 143–151.
[22] J.G. Van der Corput, Verteilungsfunktionen, Proceedings of the Koninklike Nederlands Akademie van Wetenschappen, 1935, pp. 813–821.
[23] E. Thiémard, Sur le calcul et la majoration de la discrépance à l'origine, (Thesis) Ecole Polytechnique fédérale de Lausanne, 2000.

[24] J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numer. Math. 2 (1960) 84–90.
[25] J.M. Hammersley, Monte Carlo methods for solving multivariable problems, Ann. N. Y. Acad. Sci. 86 (1960) 844–874.
[26] 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.
[27] I.M. Sobol, Uniformly distributed sequences with an additional uniform property, USSR Comput. Math. Math. Phys. 16 (1976) 236–242.
[28] J. Franco, X. Bay, D. Dupuy, B. Corre, Planification d'expériences numériques à partir du processus ponctuel de Strauss, 2008.
[29] D.J. Strauss, A model for clustering, Biometrika 62 (1975) 467–475.
[30] R. Bellman, Dynamic Programming, Princeton University Press, 1957.