Description et sélection de données en grande dimension

Aurelie Beal

Université d'Aix-Marseille

Laboratoire d'**I**nstrumentation et de **S**ciences **A**nalytiques

Ecole doctorale des Sciences Chimiques – ED250

Thèse

pour obtenir le grade de

Docteur de l'Université d'Aix-Marseille

Spécialité : *Sciences chimiques*

DESCRIPTION ET SELECTION DE DONNEES EN GRANDE DIMENSION

présentée par

AURELIE BEAL

Date de soutenance : 24 Février 2015

Pr. Thierry BASTOGNE	Rapporteur
Dr. Claire BORDES	Rapporteur
Pr. Laurence CHARLES	Examinateur
M. Claude-Alain SABY	Examinateur
Pr. Michelle SERGENT	Directrice de thèse
Dr. Magalie CLAEYS-BRUNO	Co-directrice de thèse

Remerciements

Il y a trois ans, je commençais cette thèse et aujourd'hui j'éprouve une sincère gratitude envers tous ceux qui ont participé à ce travail et je tiens ici à les remercier.

Je voudrais tout d'abord remercier grandement **Michelle Sergent** ma directrice de thèse et **Magalie Claeys-Bruno** ma co-directrice de thèse pour m'avoir fait confiance il y a trois ans. Je suis ravie d'avoir travaillé en votre compagnie car en plus de vos compétences, de votre rigueur intellectuelle, de votre dynamisme et de votre efficacité, vous avez toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse. Merci de votre confiance et de m'avoir permis de réaliser ce travail.

Thierry Bastogne et **Claire Bordes** m'ont fait l'honneur de juger ce travail en étant rapporteurs, je vous remercie pour le temps consacré à la lecture du manuscrit ainsi que pour les commentaires m'ayant permis de l'améliorer.

Je tiens à remercier **Laurence Charles** pour avoir accepté de participer à mon jury de thèse.

Claude-Alain Saby a été non seulement d'être mon parrain de thèse et a été le premier à me présenter l'ACC au tout début de ma thèse. Pour cela je vous remercie d'avoir accepté d'être dans le jury.

Mais tout ce travail ne serait rien sans le soutien de nombreuses autres personnes. Encore un grand merci à ma **grande chef** et ma **petite chef**, sans vous ce travail n'aurait pas pu aboutir, merci de m'avoir permis de réaliser cette thèse à vos côtés, et merci de m'avoir conseillée, aidée et soutenue à tout moment.

J'adresse également mes remerciement à Monsieur **Phan-Tan-Luu** qui a toujours été présent et volontaire pour échanger des idées même si certaines fois les débats ont été un peu tumultueux mais c'est sans rancunes et à compter de ce jour je lève l'interdiction de votre visite au laboratoire le vendredi après-midi.

Je remercie aussi toutes les personnes qui ont accepté de partager des jeux de données et qui m'ont ainsi permis de tester mes méthodes sur des cas d'application réels.

Merci, à toutes les autres membres du laboratoire **LISA** qui ont toujours eu un petit mot gentil en entrant dans mon bureau, merci à toute l'**équipe METICA** mais également à **Monsieur Yves**, **Aurika** et **Jacques**, pour leur extrême gentillesse et ne vous inquiétez pas je tiens toujours mon tableur à jour pour le compte des cafés... Merci à **Ghislaine** pour sa bonne humeur, son sourire et nos danses endiablées du vendredi...

A titre plus personnel, je tiens à remercier tout mon entourage ”hors laboratoire”. Je remercie bien évidemment mes **parents**, mes frères **Nicolas** et **Clément**, mais aussi ma **grand-mère**, **Christelle**, tout le reste de **ma famille en Auvergne**, **en Alsace** et **au Canada** ainsi que ma **belle-famille**. Un grand merci à tous pour votre présence et votre soutien inconditionnel.

Un grand merci aussi à tous mes amis : **Sophie**, **Nicolas**, **Barbara**, **Benoît**, **Laurent**, **Céline**, **Sylvain**, **Enora**, ... pour tous les bons moments que nous passons ensemble, pour toutes ces parties de rigolade, et ce n'est pas fini ! ! ! J'espère que nous partagerons encore pendant très longtemps des moments inoubliables.

Enfin, mes derniers remerciements sont pour les deux personnes qui chaque jour font mon bonheur et qui me rendent fière de la famille que nous formons, merci à **Éric** et **Elena**. Chaque jour vous êtes là pour moi, vous m'accompagnez et vous me soutenez, vous me faites rire même dans les moments difficiles, vous me témoignez votre amour et pour tout cela, je vous remercie du fond du cœur.

Table des matières

Introduction générale **11**

Partie I : Présentation des méthodes **17**

Chapitre 1 : Évaluation de la qualité d'une structure **21**
1.1 Critère *Mindist* . . . 21
1.2 Critère *MoyMin* . . . 21
1.3 Critère *Coverage* . . . 22
1.4 Comparaison de distributions de points selon les critères d'uniformité . . . 22
1.5 Critère *écart-type* . . . 23

Chapitre 2 : Les méthodes de visualisation des données **25**
2.1 Bibliographie des méthodes de visualisation des données en grande dimension . . . 26
2.1.1 La méthode MDS (Multidimensional scaling) . . . 26
2.1.2 Les cartes de Sammon . . . 27
2.1.3 La méthode RPM (Relational Perspective Map) . . . 28
2.1.4 La méthode Isomap (Isometric Feature Mapping) . . . 29
2.1.5 La méthode LLE (Locally Linear Embedding) . . . 30
2.1.6 Les cartes auto-organisatrices de Kohonen . . . 32
2.1.7 Les cartes GTM (Generative Topographic Mapping) . . . 34
2.1.8 Analyse en Composantes Curvilignes (ACC) . . . 34
2.1.8.1 Algorithme . . . 34
2.1.8.2 Quantification vectorielle . . . 37
2.1.8.3 Illustrations de l'ACC . . . 38
2.2 Synthèse et analyse des méthodes de visualisation des données . . . 41
2.3 Les avancées des méthodes de visualisation . . . 46
2.4 Conclusion des méthodes de visualisation des données . . . 49

Chapitre 3 : Les méthodes de sélection **51**
3.1 Les méthodes de sélection de points . . . 52
3.1.1 Méthodes de sélection de points basées sur les distances . . . 53
3.1.1.1 Algorithme de Kennard et Stone (KS) . . . 53
3.1.1.2 Algorithme DUPLEX . . . 56
3.1.1.3 Algorithme DBOD (Distance-Based Optimal Design) . . . 58

3.1.1.4 Algorithme OptiSim 60
3.1.1.5 Algorithme WSP 62
3.1.1.6 Méthode de Puchwein 65
3.1.2 Méthodes de sélection de points basées sur les clusters 66
3.1.2.1 DBSCAN (Density Based Spatial Clustering of Applications with Noise) 66
3.1.2.2 Méthode des *k*-means 68
3.1.3 Comparaison de la qualité des sous-ensembles de points sélectionnés par les différentes méthodes 69
3.1.3.1 Exemple en deux dimensions 69
3.1.3.2 Exemple en 12 dimensions 72
3.1.4 Avantages et inconvénients des méthodes de sélection de points 76
3.1.5 Les avancées des méthodes de sélection de points 78
3.1.5.1 Densification du centre du domaine 80
3.1.5.2 Densification d'une zone d'intérêt 84
3.2 Les méthodes de sélection de variables 88
3.2.1 Les méthodes B2 et B4 88
3.2.2 Le facteur d'inflation *K* 89
3.2.3 La méthode de corrélation par paires 89
3.2.4 La méthode CMC 89
3.2.5 La méthode UFS 89
3.2.6 Les méthodes de sélection pas à pas 90
3.2.7 Algorithme V-WSP 90
3.3 Conclusion sur les méthodes de sélection 93

Chapitre 4 : Le reconditionnement **95**
4.1 Élimination des amas 96
4.2 Remplissage des lacunes 101
4.3 Stratégies de reconditionnement 103
4.4 Conclusion sur le reconditionnement 109

Partie II : Cas d'étude **113**

Chapitre 1 : Les études QSAR **117**
1.1 Cas d'étude N°1 : préparation d'échantillons MALDI 119
1.2 Cas d'étude N°2 : étude de solvants 126
1.3 Cas d'étude N°3 : étude de l'excès énantiomérique 134
1.3.1 Sélection des variables pour une faible valeur de *thr* 135
1.3.2 Sélection des variables communes à toutes les solutions de V-WSP 137
1.3.3 Sélection des variables les plus représentatives par régression stepwise 138
1.4 Conclusion 141

Chapitre 2 : Analyse des données spectroscopiques **143**
2.1 Étude de fromages par infrarouge . . . 145
2.2 Étude d'une base de données constituée par des spectres infrarouges . . . 148
2.2.1 Étude des sous-ensembles de calibration et de validation . . . 148
2.2.2 Évaluation de la qualité de modèles de régression PLS . . . 149
2.2.2.1 Critères *a posteriori* pour comparer les modèles de régression PLS . . . 149
2.2.2.2 Critères pour la détection des outliers . . . 150
2.2.3 Étude de la réponse Y1 . . . 151
2.2.3.1 Construction et caractérisation des sous-ensembles de calibration et de validation . . . 151
2.2.3.2 Détermination du nombre de composantes pour la modélisation par PLS 153
2.2.3.3 Calcul des critères *a posteriori* . . . 154
2.2.3.4 Analyse des résidus en Y . . . 155
2.2.3.5 Analyse des résidus en X . . . 157
2.2.3.6 Calcul des critères *a posteriori* après suppression des outliers . . . 160
2.2.4 Étude de la réponse Y2 . . . 163
2.2.4.1 Construction et caractérisation des sous-ensembles de calibration et de validation . . . 163
2.2.4.2 Détermination du nombre de composantes pour la modélisation par PLS 165
2.2.4.3 Calcul des critères *a posteriori* . . . 166
2.2.4.4 Analyse des résidus en Y . . . 167
2.2.5 Étude de la réponse Y2 après suppression des outliers en Y . . . 170
2.2.5.1 Construction et caractérisation des sous-ensembles de calibration et de validation . . . 170
2.2.5.2 Calculs des critères *a posteriori* à partir des modèles de régression PLS 171
2.2.5.3 Analyse des résidus en X après suppression des outliers en Y . . . 172
2.2.6 Conclusion . . . 175
2.3 Conclusion de l'analyse des données spectroscopiques . . . 177

Chapitre 3 : Applications à la simulation numérique **179**
3.1 État de l'art des plans uniformes . . . 181
3.2 Exemple en 20D . . . 188
3.2.1 Réparation des plans . . . 191
3.2.1.1 Étape 1 : Élimination des amas . . . 191
3.2.1.2 Étape 2 : Remplissage des lacunes . . . 192
3.2.1.3 Étape 3 : Application de l'ACC et calcul des ratios R . . . 193
3.2.2 Conclusion . . . 193
3.3 Repliage . . . 194
3.3.1 Démarche . . . 194
3.3.2 Caractérisation des plans en dimension D . . . 195
3.3.3 Repliage . . . 197

3.3.4 Caractérisation des plans en dimension D' 198
3.3.4.1 Caractérisation des plans 10D repliés 198
3.3.4.2 Caractérisation des plans 30D repliés 201
3.3.4.3 Caractérisation des plans 50D repliés 204
3.3.5 Étude des sous-espaces en dimension D' 205
3.3.5.1 Étude de la dimension D et de la dimension D' 207
3.3.5.2 Étude de la nature du plan 208
3.3.5.3 Synthèse 208
3.3.6 Perspectives pour la réparation des plans en dimension D' 209
3.3.6.1 Première stratégie de réparation 209
3.3.6.2 Deuxième stratégie de réparation 213
3.3.7 Conclusion 215
Conclusion et perspectives **217**
Bibliographie **227**

Introduction générale

Contexte et problématique

De nos jours, un des défis majeurs dans le domaine du traitement de l'information réside dans la gestion de gros volumes de données. En effet, dans de nombreux domaines tels que la spectroscopie, les études quantitatives de relations structure-activité (QSAR) ou encore les simulations numériques de tous ordres mettant en œuvre de gros codes de calcul, l'évolution des technologies permet aujourd'hui l'acquisition d'information tant au niveau des données de sortie (output data) que des données d'entrée (input data). Cet état de fait conduit à ce que nous appellerons des "données en grande dimension". Ces données se présentent généralement sous la forme de grands tableaux pour lesquels le terme de "grande dimension" est lié soit au grand nombre de lignes lorsque les individus (ou échantillons) sont multiples, soit au grand nombre de colonnes lorsque le nombre de paramètres (ou variables d'entrée) est important.

Des domaines tels que la chimie analytique, et plus particulièrement la spectroscopie, ont vu le développement d'appareils de plus en plus performants en termes de rapidité d'acquisition et de précision de mesure, ce qui entraine immanquablement de gros volumes de données. Ce phénomène se retrouve dans d'autres applications comme les études de relations structure-activité (QSAR), l'imagerie, la génomique, ... Une des caractéristiques de ce type de données est qu'elles constituent des résultats, ce qui signifie qu'elles sont "subies", sans connaissance *a priori* ni garantie de la qualité de l'information qu'elles apportent, ce qui peut s'avérer problématique.

Un autre domaine, celui de la simulation numérique, a soulevé d'autres questionnements liés à la grande dimension. En effet, il est d'usage d'utiliser des modèles de simulation pour représenter au mieux des phénomènes réels, via des codes de calcul comme outils de prévision, d'optimisation et de décision. Mais ces codes de calcul deviennent de plus en plus lourds à gérer par la complexification toujours croissante des modèles sous-jacents, qui se traduit par un nombre de variables d'entrée sans cesse grandissant, et malgré de réelles avancées dans les performances des moyens de calcul, les temps de calcul peuvent parfois être considérables, voire rédhibitoires.

Quel que soit le domaine d'application, il peut être difficile d'apprécier la pertinence de l'information que recèle un ensemble volumineux de données. En amont de toute étude, nous devons donc nous poser la question de la qualité intrinsèque de l'information apportée par l'ensemble des points constituant la base de données, au sens large du terme. Il semble pertinent de penser qu'une répartition uniforme des points (ou des échantillons) dans l'espace des variables d'entrée est garante d'une connaissance fiable de l'espace multidimensionnel étudié. En effet, par exemple un mauvais conditionnement de la base d'apprentissage au sens d'un recouvrement non optimal de l'espace peut être problématique lors d'une étape de modélisation dans le cas de phénomènes très chaotiques. De même, dans les études QSAR, il est important de connaître la répartition des individus dans l'espace des descripteurs pour évaluer le recouvrement de l'espace. Le premier besoin se situe donc au niveau du développement de méthodes pour décrire et visualiser des données en grand nombre et dans des espaces en grande dimension. De même, cette uniformité du remplissage de l'espace demeure intéressante pour la construction de plans d'expériences et la difficulté réside alors dans le développement d'algorithmes performants et rapides en grande dimension, c'est à dire avec des dizaines, voire des centaines, de facteurs.

Néanmoins, l'évaluation d'une base de données ou la considération d'un grand nombre de variables n'écarte pas le risque de surabondance d'information qui se caractérise par des amas de points lorsque le tableau de données comporte beaucoup de lignes, ou par de fortes corrélations entre variables

lorsque celles-ci sont en nombre important. Il nous faudrait donc disposer d'outils performants pour extraire l'information la plus représentative possible de l'ensemble initial en réduisant la dimension soit par une élimination de points (lignes), soit par une élimination de variables (colonnes).

Ce sont ces différentes problématiques reliées à la gestion de la grande dimension qui sont à l'origine des travaux de recherche présentés dans ce manuscrit, car les outils à notre disposition actuellement se révèlent insuffisants.

Objectifs et axes de recherche

Il apparait que l'étude de données en grande dimension pose des problèmes spécifiques aux méthodes pour les analyser, méthodes qui ont été pour la plupart développées initialement pour des espaces de faible dimension. Aussi, serait-il utile de disposer d'un catalogue de méthodes permettant d'étudier n'importe quelle base de données, même en grande dimension, afin de pouvoir en évaluer la qualité intrinsèque, visualiser la répartition de ces données et, le cas échéant, en réduire la dimension avec la possibilité de les restructurer.

Tout d'abord, nous allons nous intéresser à la description des données en proposant des critères intrinsèques pour caractériser la qualité de la structure, au sens du positionnement des points dans l'espace multidimensionnel. Puis nous proposerons des méthodes pour visualiser les données sans utiliser des projections dans des plans de coupe mais en les projetant dans un nouvel espace en conservant au mieux l'information contenue dans l'espace initial de grande dimension. Cette description et cette visualisation peuvent mettre en évidence un mauvais conditionnement au sens d'une distribution non uniforme des données dans l'espace des variables avec des zones denses en information, appelées "amas", et/ou des zones vides, appelées "lacunes", ce qui nous a conduits à développer de nouvelles méthodes pour corriger cette perte d'uniformité.

Après la caractérisation et la visualisation, nous nous intéresserons à l'éventuelle surabondance d'information contenue dans ces données, qui nécessiterait de sélectionner judicieusement un sous-ensemble de points ou de variables représentatif. Le deuxième thème de recherche de cette thèse concernera donc les méthodes de sélection : sélection de lignes ou sélection de colonnes dans des matrices de grande dimension.

Plan de thèse

Nous avons choisi de diviser ce manuscrit en deux parties. La première sera consacrée aux méthodes que nous proposons pour apporter des réponses aux problématiques décrites ci-dessus puis, dans la seconde, nous en exposerons des cas d'application.

La **première partie** se divise en quatre chapitres et sera consacrée à la présentation de méthodes rendant plus aisé le traitement des données en grande dimension.

Dans le ***premier chapitre***, nous présenterons des critères pour évaluer la qualité d'une structure. Pour cela, nous avons choisi de ne nous intéresser qu'à la disposition des points dans l'espace des variables à partir de critères basés sur les distances euclidiennes tels que les valeurs *Mindist*, *MoyMin* et *Coverage* qui nous semblent des indicateurs pertinents pour caractériser l'uniformité d'une distribution.

Dans le ***second chapitre***, nous exposerons des méthodes pour visualiser des données dans un espace en deux dimensions sans utiliser de simples projections dans des plans de coupe. Ce sont des

méthodes qui permettent de projeter les données dans un nouvel espace en deux dimensions avec comme objectif, pour certaines, de conserver les propriétés globales, pour d'autres, de privilégier les propriétés locales ou encore de classer les données en fonction de leur similarité. Parmi ces méthodes, l'Analyse en Composantes Curvilignes (ACC) a particulièrement retenu notre attention par sa capacité à projeter les données dans un espace de faible dimension en préservant la topologie locale.

Dans le ***troisième chapitre***, nous présenterons des méthodes de sélection afin de réduire la dimensionnalité et ne conserver que les dimensions les plus représentatives de l'espace initial. Nous débuterons ce chapitre par les méthodes de sélection de points qui cherchent à extraire un échantillon de données représentatif de la population initiale. Nous comparerons ensuite ces méthodes sur des exemples simulés en 2D et en 12D en tenant compte non seulement de la qualité des sous-ensembles retenus mais aussi des temps de calcul des différents algorithmes. Parmi toutes ces méthodes, le meilleur compromis semble être l'algorithme WSP qui est facile et rapide à mettre en œuvre et qui garantit de sélectionner les points uniformément répartis dans l'espace des variables. Nous proposerons ensuite les améliorations que nous avons apportées à cet algorithme pour densifier une zone du domaine d'un plus grand intérêt ou de minimiser les conséquences du fléau de la dimension. Par la suite, nous ferons un état de l'art des méthodes de sélection de variables qui ont été développées pour sélectionner les variables les plus pertinentes en éliminant la forte corrélation qui peut exister entre celles-ci. Parmi ces méthodes, l'algorithme V-WSP a retenu notre attention par sa simplicité et ses premiers résultats prometteurs.

Dans le ***quatrième chapitre***, nous proposerons des stratégies pour reconditionner une base de données mal conditionnée laissant la possibilité de supprimer les amas de points ou de combler les zones déficientes en information.

Avec toutes ces méthodes, nous avons constitué un "catalogue" d'outils adaptés pour traiter les données en grande dimension.

Dans la **seconde partie**, nous proposerons des cas d'étude en grande dimension issus des études QSAR, de la spectroscopie ou de la simulation numérique. Selon l'objectif de l'étude et les informations requises, le traitement des données a nécessité l'utilisation d'au moins une des méthodes du "catalogue" présenté dans la première partie.

Première partie
Présentation des méthodes

Introduction

Comme nous l'avons dit précédemment, les bases de données en grande dimension peuvent être assimilées à de grandes matrices dont les lignes sont les expériences ou simulations et les colonnes sont les variables d'entrée avec leurs différentes occurrences. Travailler avec de tels tableaux a deux conséquences :

— le grand nombre de variables rend toute visualisation de la répartition des points dans l'espace multidimensionnel impossible. Seuls des espaces de projection de dimension 2 ou 3 peuvent être envisagés avec implicitement une perte de l'information,

— la surabondance d'information peut entrainer une redondance dans les lignes, avec des points très proches, ou dans les colonnes, avec de fortes corrélations entre les variables.

Par conséquent, le traitement de tableaux en grande dimension requiert des outils spécifiques, à la fois pour apprécier la répartition des données dans l'espace et les visualiser, évaluer leur qualité intrinsèque ou sélectionner des sous-ensembles de points ou de variables représentatifs.

Dans cette première partie, qui comprend quatre chapitres, nous présenterons différentes méthodes d'évaluation, de visualisation, de sélection de données et les améliorations que la grande dimension exige.

Le **premier chapitre** sera consacré à des critères qui permettent d'évaluer la qualité intrinsèque d'une structure au sens de l'uniformité de la répartition des données dans l'espace.

Dans le **second chapitre**, nous présenterons les différentes méthodes de visualisation de données, sachant que l'objectif est de projeter les données dans un espace de faible dimension (2D ou 3D) tout en conservant au mieux la topologie initiale. Nous proposerons également des adaptations et/ou des améliorations desdites méthodes, qui sont apparues nécessaires pour leur utilisation en grande dimension.

Le **troisième chapitre** compare les méthodes de sélection - sélection de points ou sélection de variables - avec les avancées afférentes. En effet, dans le cas d'une surabondance d'information, il peut être pertinent de procéder à une sélection d'un sous-ensemble de lignes ou de colonnes tout en conservant la majeure partie de l'information. Dans le premier cas, lorsque les données sont nombreuses (lignes), nous proposons d'utiliser des méthodes de sélection de points afin de ne considérer qu'un sous-ensemble judicieusement choisi et représentatif. Dans le cas où le nombre de variables est grand (colonnes), nous proposons de ne retenir que les variables les moins corrélées.

Ces méthodes de sélection de points et/ou de variables conduisent à des sous-ensembles et donc à des bases de données de taille réduite dont il faudra évaluer la qualité. Les critères intrinsèques d'évaluation de la qualité d'une répartition de points ou les méthodes de visualisation pourront alors être utilisés pour mettre en évidence entre autres la non-uniformité de la distribution des données. Cette perte d'uniformité peut se caractériser par une accumulation de données, que nous appellerons "amas",

en certaines zones de l'espace ou par l'absence de données, que l'on désignera par le terme de "lacunes". Ce mauvais conditionnement peut nécessiter une étape d'homogénéisation de la distribution des données qui permettra, le cas échéant, de supprimer les amas et de combler les lacunes.

Le **quatrième et dernier chapitre** porte donc sur le reconditionnement d'une structure et présente les méthodes que nous avons développées pour "réparer" une base de données quand celles-ci ne sont pas uniformément réparties dans l'espace des variables et cela, quelle que soit la dimension.

Chapitre 1

Évaluation de la qualité d'une structure

L'évaluation de la qualité de la structure d'un ensemble de points issu d'une base de données ou d'un plan d'expériences nécessite l'utilisation de critères quantitatifs. Il existe de nombreux critères qui permettent, entre autres, d'évaluer l'uniformité d'une répartition de points tels que la discrépance [1, 2], le critère Audze-Eglais [3, 4], les critères basés sur les distances, ... En effet, la propriété d'uniformité semble garante d'une bonne représentativité. Dans le cas de l'étude de structures en grande dimension, nous n'avons retenu que les critères basés sur les distances euclidiennes entre points, qui même en grande dimension sont faciles et rapides à calculer.

1.1 Critère *Mindist*

Le critère de distance Maximin, noté *Mindist* [5, 6, 7], définit la plus petite distance entre deux points quelconques d'une distribution (équation (1.1)) :

$$Mindist = \min_{x_i \in X} \min_{x_{k,k \neq i}} dist(x_i, x_k) \tag{1.1}$$

avec $X = \{x_1, x_2, ..., x_N\} \subset [0,1]^D$ un ensemble de N points en D dimensions.

Une valeur élevée du critère *Mindist* sera synonyme d'une bonne répartition des points dans l'espace des variables. *A contrario*, si une distribution de points présente une valeur *Mindist* faible, cela signifie qu'il y aura au moins deux points très proches ce qui révèle une zone plus dense dans le domaine que nous pourrons qualifier d'amas. En présence d'amas, la répartition des points dans le domaine ne sera plus idéale au sens de l'uniformité.

1.2 Critère *MoyMin*

Le critère *MoyMin* (équation (1.2)) définit la moyenne des distances minimales, et renseigne sur l'ensemble des distances minimales entre les points :

$$MoyMin = \frac{1}{N} \sum_{i=1}^{N} \min_{k \neq i} dist(x_i, x_k) \tag{1.2}$$

Un ensemble de N points pourra être qualifié d'uniforme si la valeur *MoyMin* est grande et proche de la valeur *Mindist*.

1.3 Critère *Coverage*

Le critère *Coverage* proposé par Gunzburger [8], mesure le recouvrement de l'espace par les points et permet de quantifier l'homogénéité de la répartition des points (équation (1.3)) :

$$Cov = \frac{1}{\bar{\gamma}} \left(\frac{1}{N} \sum_{i=1}^{N} (\bar{\gamma} - \gamma_i)^2 \right)^{1/2} \tag{1.3}$$

avec $\gamma_i = \min_{k \neq i} dist(x_i, x_k)$ et $\bar{\gamma} = \frac{1}{N} \sum_{i=1}^{N} \gamma_i$

Cette mesure renseigne sur l'homogénéité des distances minimales entre les points du plan et traduit la dispersion des distances minimales. Ainsi, lorsqu'un ensemble de points présente une valeur *Coverage* élevée, cela signifie qu'il existe une forte hétérogénéité dans la répartition des distances minimales. Certaines zones de l'espace seront alors plus densément remplies que d'autres.

1.4 Comparaison de distributions de points selon les critères d'uniformité

Les critères d'uniformité tels que *Mindist*, *MoyMin* et *Coverage* permettent de comparer des ensembles de points construits dans la même dimension, pour un même nombre de points et ainsi de classer des distributions de points en fonction du type de répartition dans l'espace (quasi-périodique, amas, gradient, aléatoire, ...). Plus précisément, nous proposons d'utiliser les représentations graphiques des valeurs $Coverage = f(Mindist)$ et $Coverage = f(MoyMin)$ pour comparer la qualité intrinsèque de différentes distributions de points. Dans ce repère la zone idéale est définie par des valeurs *Mindist* élevées et proches des valeurs *MoyMin*, avec des valeurs *Coverage* faibles. Pour ces représentations graphiques, il est important de préciser que nous ne pouvons pas mettre *a priori* de valeurs quantitatives sur les axes car elles dépendent à la fois du nombre de points N et de la dimension D.

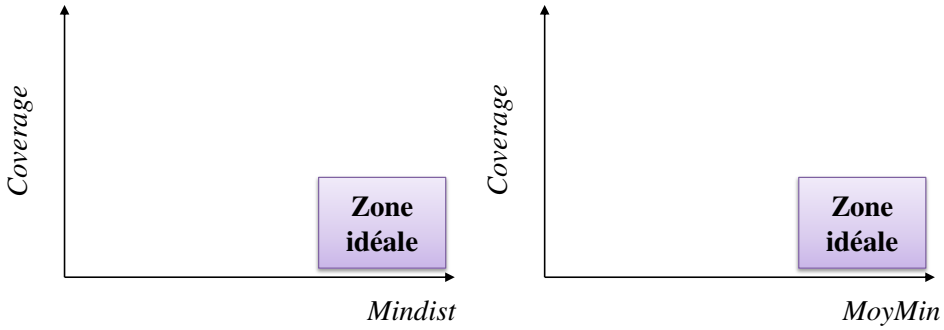
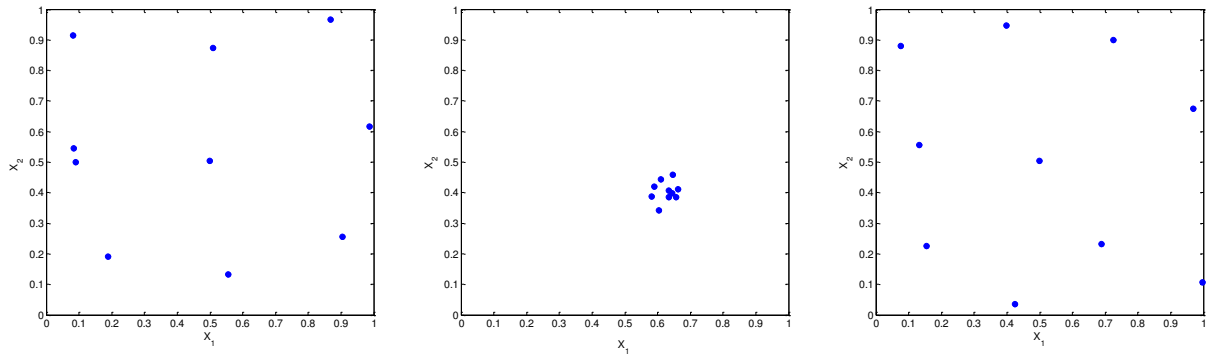


FIGURE 1.1 – Représentation graphique de la zone idéale selon les critères $Coverage = f(Mindist)$ et $Coverage = f(MoyMin)$. Pour une valeur N fixée, la zone idéale présente une valeur élevée du critère *Mindist* qui est proche du critère *MoyMin,* garantissant une distance suffisante entre tous les points et une faible valeur *Coverage* caractérisant une homogénéité des distances minimales.

Ainsi, pour une distribution de N points dans un espace en D dimensions, l'analyse des valeurs *Mindist*, *MoyMin* et *Coverage* renseigne sur la qualité de cette distribution au sens de l'uniformité. Par exemple, une valeur *Mindist* faible et une valeur *MoyMin* élevée, signifient qu'au moins deux points sont très proches mais que les autres points sont bien répartis dans l'espace des variables. Si les valeurs *Mindist* et *MoyMin* sont équivalentes et faibles, cela signifie que les plus petites distances entre points

sont équivalentes c'est-à-dire que tous les points sont proches et forment un seul amas. *A contrario*, si une distribution présente des valeurs *Mindist* et *MoyMin* élevées, cette distribution sera plus uniforme (figure 1.2).

a. b. c.



Mindist faible, *MoyMin* grand | *Mindist* $\sim$ *MoyMin* et faibles | *Mindist* $\sim$ *MoyMin* et grands

FIGURE 1.2 – Comparaison de trois distributions de points. **a)** distribution uniforme avec deux points très proches, **b)** tous les points sont proches et constituent un seul amas, **c)** distribution uniforme.

1.5 Critère *écart-type*

Les critères *Mindist*, *MoyMin* et *Coverage* permettent de comparer des ensembles de points avec le même nombre de points. Il nous semble intéressant d'ajouter un critère complémentaire permettant de comparer des ensembles non équivalents en nombre de points. Nous ajoutons donc aux critères précédents, l'*écart-type* σ (équation (1.4)) :

$$\sigma = \left(\frac{1}{N} \sum_{i=1}^{N} (\bar{\gamma} - \gamma_i)^2 \right)^{1/2} \tag{1.4}$$

Contrairement au *Coverage*, il a été montré que l'*écart-type* est indépendant du nombre de points constituant l'ensemble considéré [9]. Ainsi, la mesure de l'*écart-type* permet de comparer des ensembles n'ayant pas le même nombre de points.

Chapitre 2

Les méthodes de visualisation des données

Quels que soient les domaines d'application, il est courant de chercher à visualiser des données en grande dimension (grand nombre de paramètres ou de variables). Or, s'il est aisé de visualiser les données en deux ou trois dimensions, l'exploration des données en grande dimension est beaucoup moins facile.

Pour pallier cette difficulté, nous faisons alors appel à des techniques de réduction de dimensionnalité qui permettent, à partir de D variables originales, de construire p nouvelles variables qui contiennent la plus grande partie de l'information initiale. C'est en effet la solution la plus naturelle : la dimension est trop grande alors réduisons-la !

L'enjeu est donc d'identifier les variables ou les combinaisons de variables qui sont les plus pertinentes. Ces méthodes peuvent se baser sur des projections linéaires (l'analyse en composantes principales [10]) ou non-linéaires (les méthodes de "Multidimensional Scaling" [11], l'analyse en composantes curvilignes [12] ...). D'une manière générale, la projection des données de l'espace original en D dimensions vers l'espace cible de dimension p ($p < D$) repose sur un critère que l'on cherchera à minimiser.

Il existe de nombreuses méthodes de réduction de la dimensionnalité qui se différencient principalement par la fonction à minimiser. Nous présenterons certaines de ces méthodes de projection non linéaire dont le but explicite est de préserver "au mieux" le voisinage des données :

- la méthode MDS (Multidimensional Scaling),
- les cartes de Sammon,
- la méthode RPM (Relational Perspective Map),
- la méthode Isomap (Isometric Feature Mapping),
- la méthode LLE (Locally Linear Embedding),
- les cartes auto-organisatrices de Kohonen,
- la méthode GTM (Generative Topographic Mapping),
- l'Analyse en Composantes Curvilignes.

Après avoir décrit les méthodes pré-citées, nous effectuerons une analyse et une synthèse de ces méthodes en listant leurs avantages et leurs inconvénients. Parmi ces méthodes, l'ACC a particulièrement retenu notre attention, car elle semble répondre à nos besoins, à savoir conserver les faibles distances entre points dans l'espace de projection. Néanmoins une méthode de visualisation ne peut apporter qu'une représentation graphique des données dans un espace de projection ce qui peut être insuffisant

lors de l'interprétation de ces données. C'est pour cela que nous proposerons dans une dernière partie de nouveaux outils qui s'appliqueront sur les données projetées et permettront d'en compléter leur interprétation.

2.1 Bibliographie des méthodes de visualisation des données en grande dimension

Afin de comparer les méthodes de réduction de dimensionnalité, nous avons choisi de les appliquer sur un exemple proposé dans la littérature décrivant une "surface enroulée" ou swissroll dans un espace à trois dimensions et constitué de 2000 points (figure 1.3).

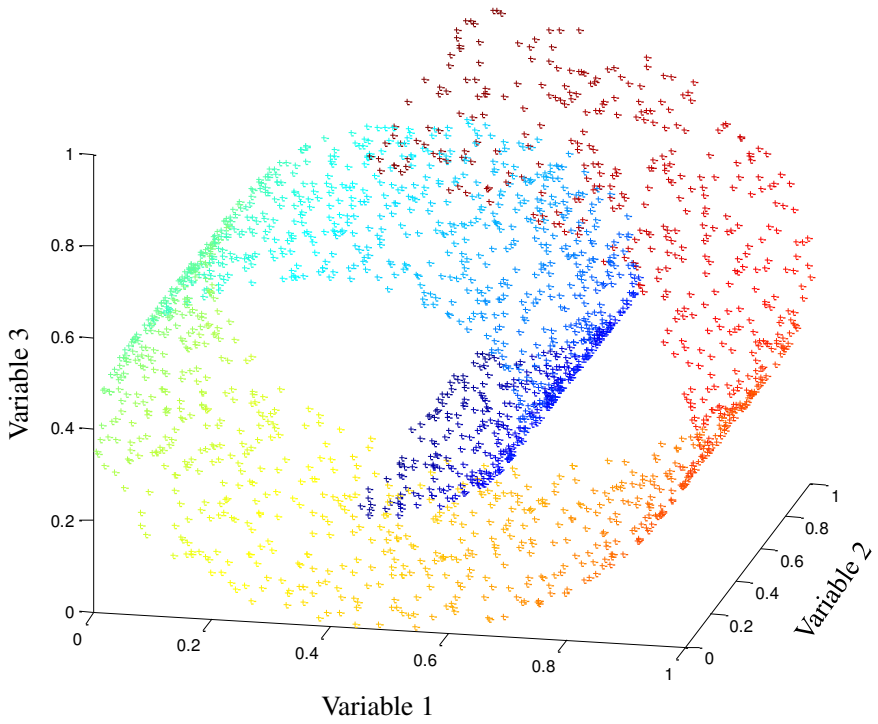


FIGURE 1.3 – Swissroll en trois dimensions et 2000 points.

Dans la suite de ce document, nous noterons X_{ij} la distance entre un point x_i et x_j dans l'espace initial en D dimensions et Y_{ij} la distance entre les images correspondantes y_i et y_j dans l'espace de projection en p dimensions.

2.1.1 La méthode MDS (Multidimensional scaling)

On désigne sous le terme de positionnement multidimensionnel ou MDS [11, 13] un ensemble de techniques non linéaires permettant de représenter sur une carte en p dimensions, N points en D dimensions (avec $p < D$). Cette réduction de dimensionnalité est effectuée en utilisant les informations relatives à la "similarité" ou "dissimilarité" entre chaque couple de points. L'objectif de la méthode MDS est de minimiser une fonction d'erreur (équation (2.1)) afin de faire correspondre au mieux les distances de sortie aux distances d'entrée.

$$E = \sum_{i<j} (X_{ij} - Y_{ij})^2 \tag{2.1}$$

La méthode MDS est souvent utilisée pour la visualisation de données, par exemple en imagerie par résonance magnétique [14] ou la modélisation moléculaire [15].

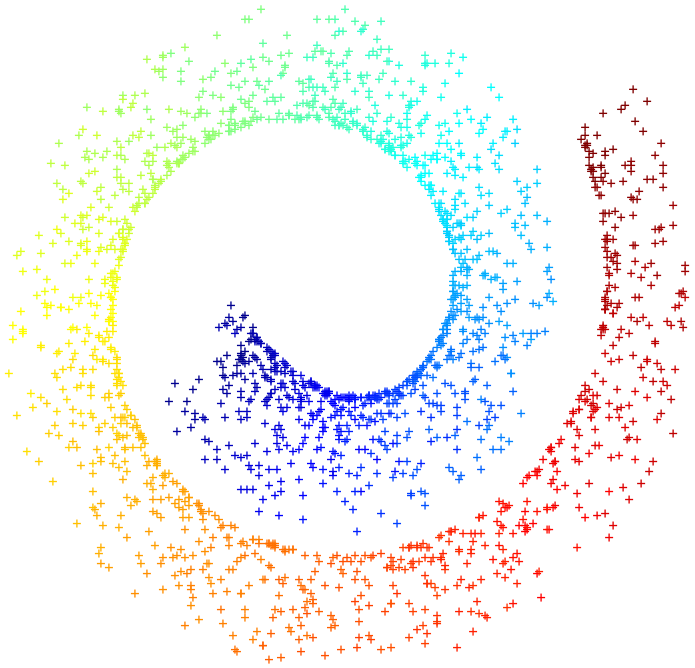


FIGURE 1.4 – Visualisation du swissroll dans un espace en deux dimensions par la méthode **MDS** métrique.

L'utilisation de la méthode MDS métrique sur l'exemple du swissroll, permet de visualiser ces données dans un espace en deux dimensions (figure 1.4). La projection conserve la géométrie initiale mais la courbure interne présente une forte densité de points entrainant ainsi une perte d'information sur la distance entre les points dans l'espace initial.

Une variante non métrique de la méthode MDS a été proposée par Shepard et Kruskal [11] dont l'objectif est de considérer la proximité entre points comme une fonction monotone des distances. A la différence d'un modèle métrique qui impose la condition que les données sont proportionnelles aux distances, un modèle non-métrique suppose que les données sont liées à une distance selon une fonction f monotone telle que (équation (2.2)) :

$$X_{ij} < X_{hk} \Rightarrow f(X_{ij}) < f(X_{hk}) \,, \forall i, j, h, k \in X \tag{2.2}$$

2.1.2 Les cartes de Sammon

La méthode des cartes de Sammon [16] est un algorithme non linéaire qui est lié à la méthode MDS. Le but de cet algorithme est de projeter N données issues d'un espace en grande dimension D vers un espace de dimension inférieure p. Durant la projection, l'algorithme de Sammon tend à conserver les distances entre tous les points du jeu de données afin de préserver au mieux la structure initiale.

La projection des N points nécessite de minimiser une fonction d'erreur définie par la relation suivante (équation (2.3)) :

$$E = \frac{1}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} X_{ij}} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{(X_{ij}-Y_{ij})^2}{X_{ij}} \tag{2.3}$$

avec X_{ij} la distance entre un point x_i et x_j dans l'espace initial en D dimensions et Y_{ij} la distance entre les images correspondantes y_i et y_j dans l'espace de projection en p dimensions.

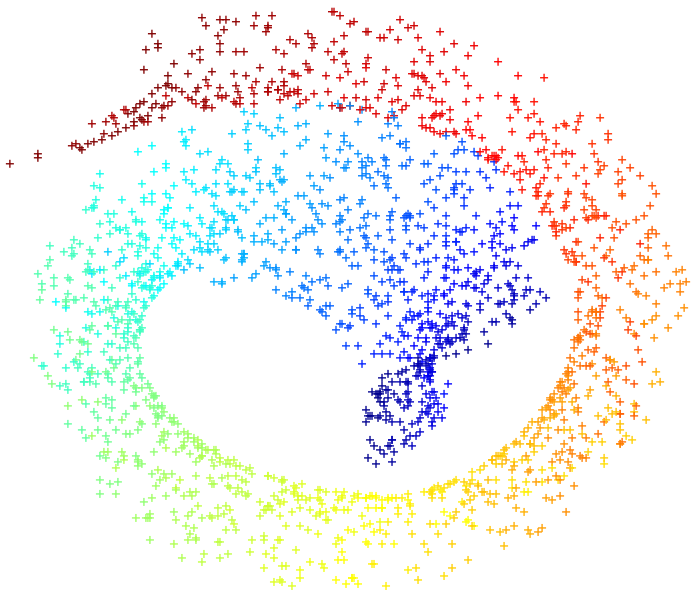


FIGURE 1.5 – Visualisation du swissroll dans un espace en deux dimensions par une **carte de Sammon**.

L'utilisation de cette méthode sur l'exemple du swissroll (figure 1.5) montre une superposition des points. Dans ce cas, cette méthode ne garantit donc pas la conservation des distances et conduit à une interprétation erronée de la disposition des points dans l'espace.

2.1.3 La méthode RPM (Relational Perspective Map)

La méthode RPM a été mise au point par Li [17]. L'objectif est de représenter sur une carte en deux dimensions les points initiaux de telle sorte que la distance euclidienne entre les points projetés (notée Y_{ij}) soit la plus proche de la distance entre ces points dans l'espace initial (notée X_{ij}). L'algorithme comporte deux étapes : la première consiste à cartographier les données sur une surface torique et la deuxième à déplier horizontalement et verticalement cette surface de telle sorte que les distances soient conservées (figure 1.6). Pour trouver la meilleure carte, l'algorithme RPM minimise la fonction énergie suivante (équation (2.4)) :

$$E_{p_r} = \sum_{i<j} \frac{X_{ij}}{p_r Y_{ij}^{p_r}} \tag{2.4}$$

avec $E_0 = -\sum_{i<j} X_{ij} ln(Y_{ij})$ et p_r, un paramètre de l'algorithme appelé rigidité.

L'algorithme RPM utilise la méthode d'optimisation par descente de gradient pour trouver une configuration avec une énergie minimale. Le paramètre de rigidité p_r, qui est normalement compris entre -1 et $+1$, modifie la structure de l'énergie de façon globale, de telle sorte que les cartes RPM présentent des caractéristiques différentes.

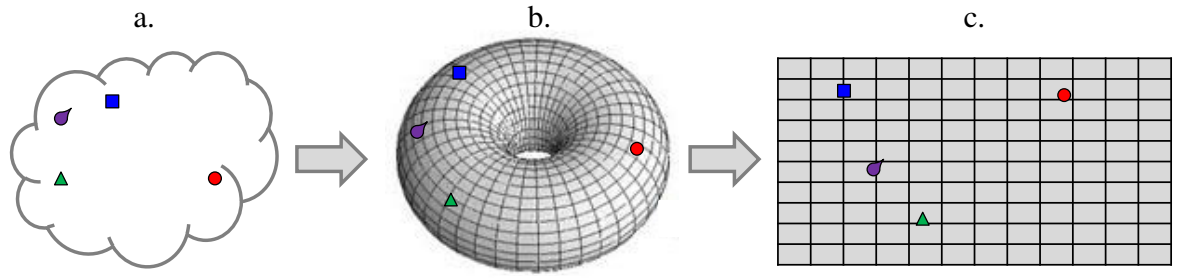


FIGURE 1.6 – Méthode **RPM** (d'après [17]). **a)** Jeu de données décrit par 4 points dans un espace en grande dimension. **b)** 4 points projetés sur une surface torique. **c)** Carte RPM.

D'un point de vue physique, l'algorithme RPM simule un système à particules, lesquelles sont soumises à des forces répulsives et nous cherchons à minimiser l'énergie potentielle. Les forces de répulsion entre deux points projetés sont proportionnelles à leur distance relationnelle et les données qui sont très éloignées dans l'espace initial sont disposées de part et d'autre de la surface torique. La figure 1.6 illustre la méthode RPM sur un jeu de données en grande dimension constitué par 4 points. Ces 4 points seront d'abord projetés sur la surface torique, laquelle sera ensuite dépliée horizontalement et verticalement pour obtenir la carte RPM des 4 points. Nous observons que le point représenté en rouge qui était initialement éloigné des trois autres points reste isolé sur la carte RPM.

Cette méthode n'est pas appliquée sur le swissroll car l'outil de traitement proposé par l'auteur n'est plus disponible.

2.1.4 La méthode Isomap (Isometric Feature Mapping)

La méthode Isomap [14] est une méthode non-linéaire de réduction de dimensionnalité. L'idée de base de la méthode Isomap est de surmonter les limitations de la métrique traditionnelle de la méthode MDS, en la remplaçant par une métrique géodésique. En effet, la méthode MDS rencontre des difficultés quand on veut projeter des données fortement non linéaires comme une spirale par exemple.

L'objectif de la méthode Isomap est donc de trouver la carte qui préserve globalement la géométrie non linéaire des données en considérant des distances géodésiques (ou curvilignes). La figure 1.7 illustre la mesure de la distance euclidienne et de la distance géodésique entre deux points. Ainsi, la distance euclidienne peut se décrire comme la plus courte distance entre deux points alors que la distance géodésique est la plus courte distance entre deux points en suivant la surface formée par les données. La méthode Isomap estime la distance géodésique [18, 19] de la façon suivante : dans un premier temps, on calcule le voisinage de chaque point en considérant, soit les k plus proches voisins, soit un ensemble de points situés à une distance inférieure à λ. Une fois le voisinage connu, un graphe de voisinage est construit en reliant tous les voisins. Chaque arête du graphe est ensuite pondérée par la distance euclidienne [18] entre les points constituant cette arête. Enfin, la distance géodésique entre deux points est approximativement la somme des longueurs des arêtes du plus court chemin entre ces deux points. En pratique ce trajet est calculé par l'algorithme de Dijkstra [20, 21].

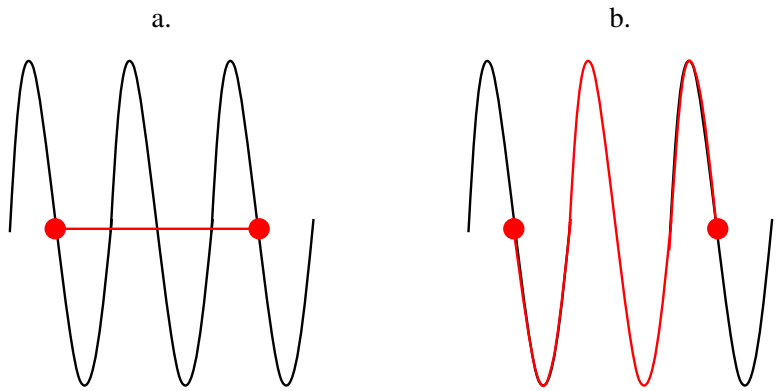


FIGURE 1.7 – **a)** Distance euclidienne. **b)** Distance géodésique

Algorithme 2.1 Algorithme Isomap

— Création d'un graphe de voisinage pour toutes les données en connectant les points i et j tels que i soit l'un des plus proches voisins de j (ou si la distance entre i et j est inférieure à λ).
— Calculer le plus court chemin géodésique entre les points sur le graphe
— Projeter les points en appliquant une méthode MDS classique pour le chemin à l'étape précédente.

L'algorithme Isomap nécessite de définir les paramètres suivants : k, le nombre de voisins ou λ, la distance du voisinage. Le choix de ces paramètres, conduit à des résultats différents comme nous pouvons le constater sur la figure 1.8 qui représente les données obtenues par la méthode Isomap en considérant 5, 7 et 10 voisins.

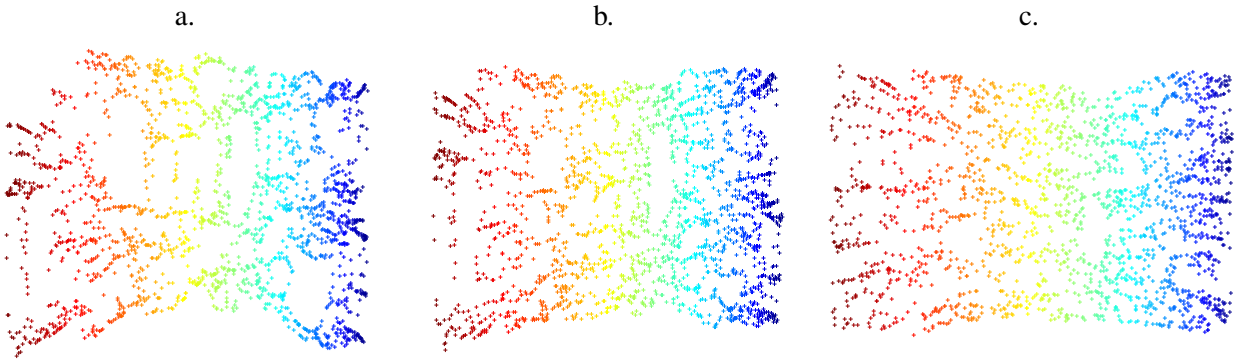


FIGURE 1.8 – Visualisation du swissroll dans un espace en deux dimensions par la méthode **Isomap** en fixant k, le nombre de voisins. **a)** $k = 5$, **b)** $k = 7$, **c)** $k = 10$.

2.1.5 La méthode LLE (Locally Linear Embedding)

La méthode LLE [22, 23] tente de résoudre le même problème que Isomap par une approche alternative. Chaque point est ici caractérisé par sa reconstruction par ses plus proches voisins. Si le nombre de points est suffisamment grand, nous pouvons supposer que chaque point et ses plus proches voisins sont approximativement situés sur une partie localement linéaire. Cette géométrie locale peut être caractérisée par des coefficients linéaires de reconstruction de chaque point à partir de ses voisins.

L'algorithme débute par l'identification des voisins de chaque point. Comme pour Isomap, nous pouvons choisir les k plus proches voisins ou bien sélectionner tous les points dans un voisinage de taille λ du point x_i. Nous pouvons mesurer l'erreur de reconstruction d'un point par ses voisins suivant la relation (équation (2.5)) :

$$erreur(W) = \sum_i \left\| x_i - \sum_j W_{ij} x_j \right\|^2 \tag{2.5}$$

où W_{ij} quantifie la contribution du point x_j dans la reconstruction de x_i.

Afin d'estimer W_{ij}, on minimise cette fonction avec deux contraintes. Tout d'abord chaque x_i n'est reconstruit qu'à partir de ses plus proches voisins, de cette façon $W_{ij} = 0$ si x_j n'est pas voisin de x_i. La seconde contrainte consiste à exiger que $\sum W_{ij} = 1$; ainsi la reconstruction d'un point à partir de ses voisins est inchangée pour toute rotation, tout changement d'échelle ou translation de ce point et de ses voisins. La dernière étape de l'algorithme LLE consiste à trouver une représentation $Y = \{y_1, y_2, ..., y_N\}$ des données initiales $X = \{x_1, x_2, ..., x_N\}$ dans un espace de dimension inférieure. Ceci est réalisé en minimisant la fonction de coût $\Phi(Y)$ (équation (2.6)).

$$\Phi(Y) = \sum_i \left\| y_i - \sum_j W_{ij} y_j \right\|^2 \tag{2.6}$$

Algorithme 2.2 Algorithme LLE

— Recherche des k plus proches voisins du point x_i.
— Calcul des poids W_{ij} qui reconstruisent chaque point x_i à partir de ses voisins.
— Projection des vecteurs en minimisant la fonction de coût $\Phi(Y)$.

L'utilisation de la méthode LLE sur l'exemple du swissroll conduit à des résultats différents selon la valeur du paramètre k désignant le nombre de voisins (figure 1.9). En effet, nous observons qu'à partir de $k = 10$, le swissroll est déroulé mais avec une perte d'information sur les distances entre points. Pour $k < 10$, la géométrie initiale est perdue ainsi que toute information sur les distances entre points.

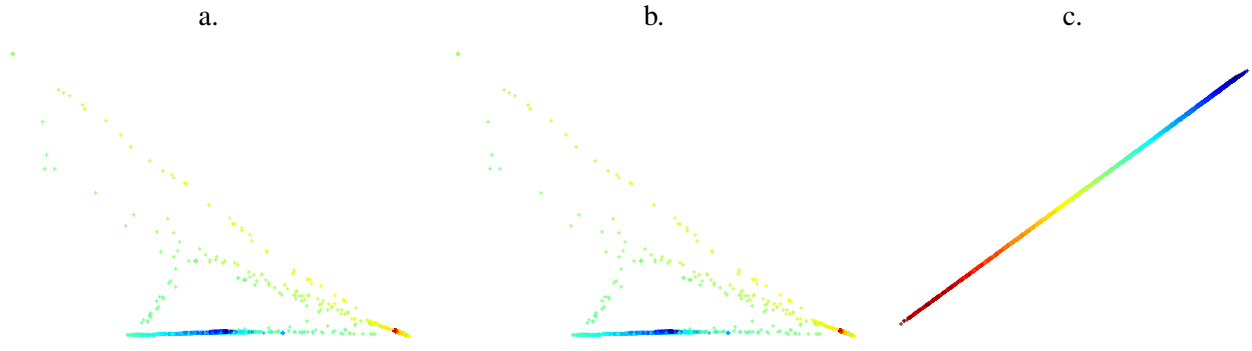


FIGURE 1.9 – Visualisation du swissroll dans un espace en deux dimensions par la méthode **LLE** en fonction du nombre de voisins k. **a)** $k = 5$, **b)** $k = 7$, **c)** $k = 10$.

2.1.6 Les cartes auto-organisatrices de Kohonen

Les cartes auto-organisatrices de Kohonen notées SOM (Self-Organized Mapping) [24, 25] reposent sur un algorithme de classification. Ces cartes sont des réseaux de neurones non supervisés qui cherchent à regrouper les données en classes tout en respectant la topologie de l'espace initial. Pour ce faire, il est nécessaire de définir *a priori* une notion de voisinage entre classes afin que des observations voisines dans l'espace des données appartiennent après classement au même groupe ou à des classes voisines. Pour cela, l'algorithme de Kohonen cherche à projeter les données dans un espace de faible dimension tout en cherchant à reproduire au mieux les corrélations existant entre les données initiales. Les cartes de Kohonen sont constituées d'une première couche qui sert uniquement à la présentation des données à classer, c'est la phase d'apprentissage, et la seconde couche appelée couche d'adaptation est formée d'une grille régulière dont chaque nœud est occupé par un neurone. Les nœuds de la carte sont disposés géométriquement selon une topologie fixée *a priori* qui peut être rectangulaire ou hexagonale (figure 1.10) imposant ainsi une notion de voisinage et de distances entre les neurones.

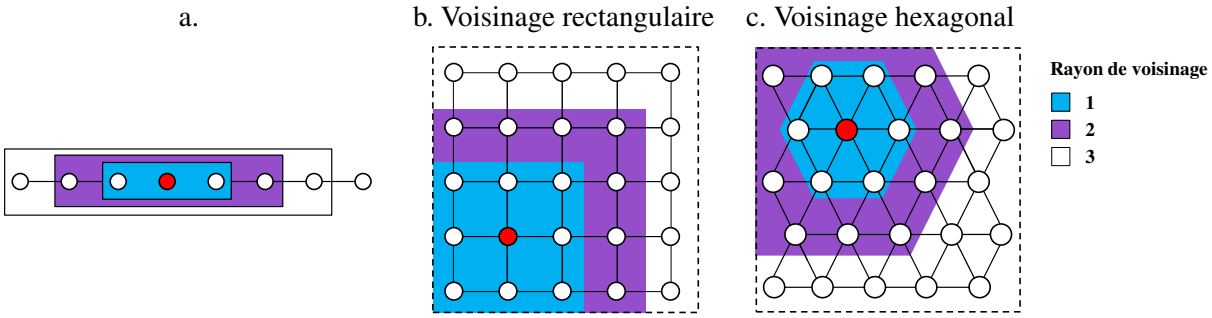


FIGURE 1.10 – Représentation du voisinage d'un neurone "gagnant" (représenté en rouge) d'une carte de Kohonen. Les cartes peuvent être **a)** unidimensionnelle ou bidimensionnelle avec un voisinage **b)** rectangulaire ou **c)** hexagonal.

Chaque élément de l'espace initial, appelé prototype, est associé à tous les neurones. L'objectif de la phase d'apprentissage pour les SOM est de déplacer certains prototypes vers les données d'apprentissage. Pour ce faire, les données sont successivement présentées à tous les neurones qui composent la carte de Kohonen. Le neurone pour lequel le prototype est le plus proche de la donnée présentée est nommé "neurone gagnant" (et le prototype associé, le "prototype gagnant"). À partir de ce neurone gagnant, un ensemble de neurones faisant partie de son voisinage est sélectionné. Nous rapprochons ensuite linéairement les prototypes des neurones de ce voisinage ainsi que le prototype gagnant vers la donnée d'apprentissage. Ces actions sont alors répétées jusqu'à ce que les déplacements des prototypes soient de faible amplitude voire quasiment nuls.

L'algorithme des cartes de Kohonen peut se résumer ainsi (Algorithme 2.3) :

Algorithme 2.3 Algorithme de Kohonen

Soit $X = \{x_1, x_2, ..., x_N\}$ les données d'apprentissage
Initialisation aléatoire du vecteur des poids W

POUR une itération t

— Choisir aléatoirement un élément $x_t \in X$

— Trouver le neurone gagnant noté $i_{gagnant}$ tel que la distance entre x_t et $w_{i_{gagnant}}$ soit minimale

$$i_{gagnant} = argmin_i(d(x_t, w_i))$$

avec d la distance euclidienne dans l'espace d'entrée et w_i la valeur du poids associé au neurone i

— Recherche du voisinage du neurone gagnant selon la règle suivante :

$$f_{i_{gagnant}}(i,t) = f(d(i, i_{gagnant}), t)$$

avec f la fonction de voisinage

— Pour chaque x_t, le neurone gagnant ainsi que les neurones situés dans son voisinage auront leur vecteur poids modifié :

$$w_i(t+1) = w_i(t) + \mu f(w_i(t) - x_t)$$

avec μ le pas d'apprentissage

FIN

L'algorithme de Kohonen nécessite de définir une valeur du pas d'apprentissage μ qui décroit au cours du temps et permet de contrôler la vitesse d'apprentissage. Si μ est trop petit, le modèle ne s'adapte pas assez aux données, si μ est trop grand il y a un risque d'instabilité du modèle. La fonction de voisinage f est une fonction continue généralement de forme gaussienne.

Les cartes de Kohonen sont utilisées pour réduire la dimension des données mais elles présentent un problème majeur non résolu qui nécessite de définir *a priori* la forme de la carte. Le plus souvent, il s'agit d'une grille à maille rectangulaire ou hexagonale. Dans la plupart des applications réelles, la forme du sous-espace de projection est inconnue et ne peut pas être estimée car la dimension de l'espace d'entrée est grande. Faute de mieux, seule une grille carrée pourra être choisie ce qui ne peut pas être adapté à tous les cas d'étude. Ainsi, l'algorithme de Kohonen ne permet aucune flexibilité du réseau de neurones puisque celui-ci est imposé au départ que se soit par sa géométrie rigide ou le nombre de neurones envisagés.

Contrairement à la méthode MDS, SOM ne conserve pas les distances mais fournit l'information topologique des relations de voisinage entre les points. Aussi, la principale utilisation de la méthode SOM est la classification des données dans un espace réduit.

Pour cette méthode, nous ne proposons pas d'illustration sur l'exemple de swissroll car nous ne pouvons obtenir qu'une classification des données.

2.1.7 Les cartes GTM (Generative Topographic Mapping)

La méthode GTM [26] peut être considérée comme une approche probabiliste des cartes de Kohonen (section 2.1.6). Le but est de modéliser la distribution de données dans un espace de dimension réduite en prenant en compte le plus petit nombre de variables latentes. Tout comme les cartes de Kohonen dont elles sont issues, les GTM reposent sur des relations de voisinage, et de classification, mais n'imposent pas la géométrie du réseau ce qui améliore les résultats de classification. Il est cependant nécessaire de fixer le nombre de neurones au départ ce qui ne permet pas au réseau de s'organiser de la meilleure façon. Toutefois, contrairement aux SOM, les GTM possèdent l'avantage de ne pas fixer les neurones sur une grille, mais de les placer dans l'espace des variables. La classification des données est ainsi plus efficace en utilisant la méthode GTM plutôt que les SOM.

Comme la méthode SOM, les GTM sont principalement utilisées pour la visualisation des données regroupées en classes.

2.1.8 Analyse en Composantes Curvilignes (ACC)

2.1.8.1 Algorithme

Le but de l'analyse en composantes curvilignes (ACC) [12, 27] est de reproduire la topologie d'un espace initial de dimension D dans un espace de dimension inférieure p dans lequel nous souhaitons projeter l'ensemble des données. La topologie générale ne pouvant pas être reproduite, l'ACC essaie de préserver la topologie locale. Pour cela, nous considérons N neurones dont les vecteurs d'entrée $\{x_i; i = 1,...,N\}$ en dimension D quantifient la distribution d'entrée et dont les vecteurs de sortie $\{y_i; i = 1,...,N\}$ en p dimensions (avec $p < D$) devront copier la topologie des x_i. Pour ce faire, on se base sur les distances entre les x_i : $X_{ij} = d(x_i, x_j)$ avec d la distance euclidienne et les distances correspondantes en sortie : $Y_{ij} = d(y_i, y_j)$.

La figure 1.11 présente le principe de fonctionnement de l'algorithme ACC appliqué sur N données en D dimensions qui sont projetées dans un nouvel espace de dimension p, avec $p < D$.

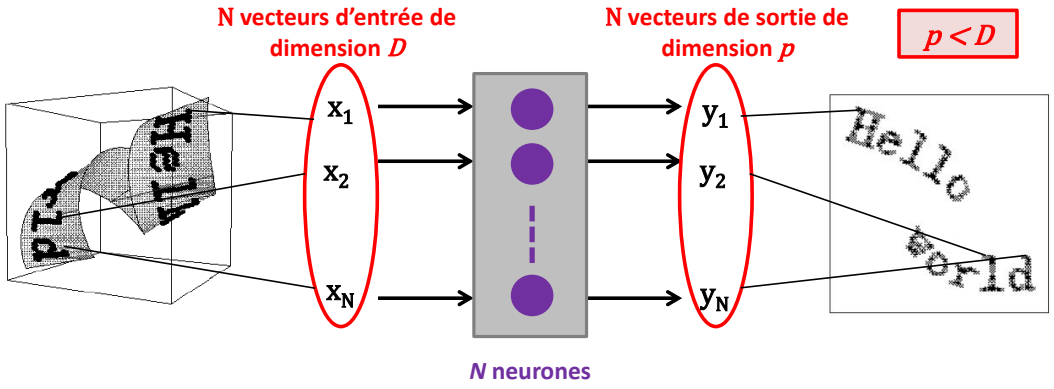


FIGURE 1.11 – Principe de fonctionnement de l'Analyse en Composantes Curvilignes.

Lors d'une projection l'objectif est de rendre équivalentes les distances Y_{ij} aux distances X_{ij}. Pour cela, on cherche à minimiser un critère E_{ACC} (équation (2.7)) qui caractérise les différences de topologie entre l'espace initial et l'espace de projection.

$$E_{ACC} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(X_{ij}-Y_{ij})^2 F_\lambda(Y_{ij}) \tag{2.7}$$

avec $F_\lambda(Y_{ij}) : \mathbb{R}_+ \to [0,1]$ une fonction monotone décroissante selon Y_{ij} qui permet de conserver la topologie locale. Cette fonction $F_\lambda(Y_{ij})$ est appelée fonction de pondération ou fonction de coût. Demartines et Hérault (1997) [27] proposent de choisir F comme une fonction paramétrée par la valeur λ appelée distance critique ou rayon de voisinage (figure 1.12).

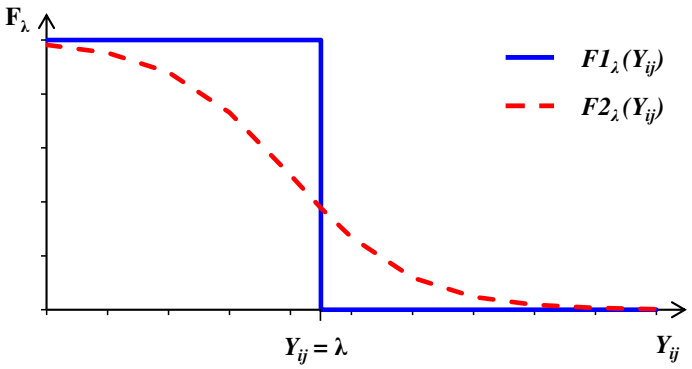


FIGURE 1.12 – Exemples de fonctions de coût visant à favoriser les faibles distances dans l'espace de sortie.

La minimisation du critère E_{ACC} peut se faire par descente de gradient qui conduit à la règle d'adaptation suivante (équation (2.8)) :

$$\Delta y_i = \alpha \sum_{j\neq i}^{N} \frac{X_{ij}-Y_{ij}}{Y_{ij}} \left[2F_\lambda(Y_{ij}) - (X_{ij}-Y_{ij})F'_\lambda(Y_{ij})\right](y_i - y_j) \tag{2.8}$$

avec α le facteur d'adaptation. L'adaptation ici est qualifiée de "passive" (figure 1.13a.) c'est à dire que le déplacement du vecteur y_i est une somme des contributions de tous les points y_j situés à une distance inférieure à λ (avec $j \neq i$). Chaque contribution est une attraction ou une répulsion de l'unité y_i vers l'unité y_j. Lorsque y_i est trop éloigné de y_j, il est rapproché de y_j, alors qu'il est éloigné dans le cas contraire. Toutefois, cette règle souffre de plusieurs défauts :

— pour chaque point y_i, il faut faire une somme sur tous les points y_j ce qui implique de lourds calculs,
— le processus d'adaptation peut tomber dans un minimum local de E_{ACC}.

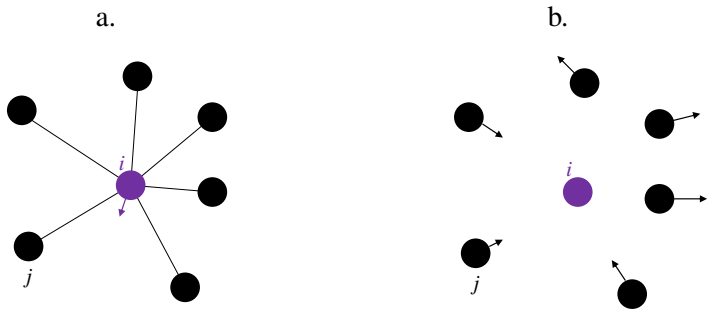


FIGURE 1.13 – Règles d'adaptation. **a)** Adaptation "passive". **b)** Adaptation "active".

Ainsi, au lieu de modifier le vecteur y_i en fonction de la somme des contributions de tous les y_j (avec $j \neq i$), l'algorithme ACC utilise une adaptation "active" qui consiste à choisir aléatoirement un point fixe y_i et de déplacer radialement tous les autres points y_j situés à une distance inférieure à λ autour du point y_i. La minimisation du critère E_{ACC} repose ainsi sur une simple descente de gradient qui donne la règle d'adaptation suivante (équation (2.9)) :

$$\Delta y_i = \alpha \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left[2F_\lambda(Y_{ij}) - (X_{ij} - Y_{ij}) F'_\lambda(Y_{ij}) \right] (y_i - y_j) \tag{2.9}$$

Les fonctions de coût les plus utilisées sont $F1$ (équation (2.10)) et $F2$ (équation (2.11)) représentées figure 1.12 :

$$F1_\lambda(Y_{ij}) = \begin{cases} 1, Y_{ij} \leq \lambda \\ 0, Y_{ij} > \lambda \end{cases} \tag{2.10}$$

$$F2_\lambda(Y_{ij}) = \frac{1}{1 + e^{(Y_{ij} - \lambda)}} \tag{2.11}$$

Pour les exemples d'application présentés ci-dessous, nous utilisons la fonction créneau $F1$ dans l'algorithme ACC. Cette fonction est intéressante car elle est positive, décroissante et sa dérivée est nulle ce qui implique la possibilité de minimiser E_{ACC} avec une descente de gradient stochastique modifiée, plus facile à calculer (équation (2.12)).

$$\Delta y_i = \alpha F_\lambda(Y_{ij}) \frac{X_{ij} - Y_{ij}}{Y_{ij}} (y_i - y_j), \forall j \neq i \tag{2.12}$$

Dans le cas où $F_\lambda(Y_{ij})$ est une fonction créneau seulement les points y_j situés à une distance inférieure à λ sont déplacés autour du point y_i.

Tout comme les autres méthodes, la méthode ACC est appliquée sur l'exemple du swissroll et la projection obtenue est représentée par la figure 1.14.

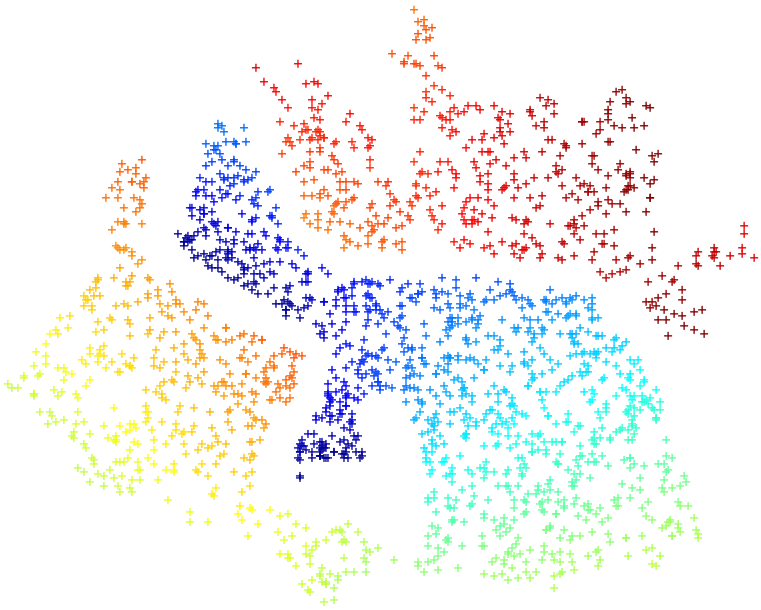


FIGURE 1.14 – Visualisation du swissroll dans un espace en deux dimensions par la méthode **ACC**.

La projection des données du swissroll par ACC conduit à une répartition de points très différente de celles obtenues par les autres méthodes. En effet, avec l'ACC la géométrie initiale n'est pas conservée puisque nous ne retrouvons pas la forme de la "spirale", mais dans le seul but de conserver au mieux les distances entre les points.

La limitation principale de l'algorithme originel de l'ACC est sa complexité de calcul. Il est évident que si l'on souhaite projeter un grand nombre de points, il est nécessaire de réduire le coût algorithmique. Pour cela, il est recommandé d'effectuer une quantification vectorielle.

2.1.8.2 Quantification vectorielle

Lorsque nous souhaitons appliquer un algorithme sur une grande base de données, le coût algorithmique est élevé. Une solution possible est d'utiliser la quantification vectorielle (notée QV) qui ne doit pas reproduire la topologie de la base de données initiale mais simplement représenter au mieux la distribution des points. La QV choisit quelques représentants pertinents dans la base de données qui sont appelés points d'ancrage ou centroïdes. La QV permet d'appliquer l'algorithme uniquement sur ces points d'ancrage représentatifs d'un groupe de points.

Ainsi la QV permet de fournir un sous-ensemble de points représentant au mieux la base de données initiale et de réduire le coût algorithmique.

Si on utilise la quantification vectorielle dans le cas de l'ACC, la stratégie devient :

— choix des points d'ancrage par QV,
— projection des points d'ancrage par ACC,
— projection de l'ensemble des points restants en fonction de leur distance par rapport aux points d'ancrage.

Les points de l'espace de sortie sont initialement disposés de façon totalement aléatoire avant d'être déplacés pour minimiser progressivement le critère de l'ACC. A chaque itération, le point d'ancrage est

fixe et tous les autres points sont déplacés tels que leur distance à ce point et uniquement à ce point se rapproche de la distance correspondante dans l'espace de départ. Seuls les points voisins du point d'ancrage sont déplacés. Cette technique évite de projeter tous les points en même temps et ainsi, de calculer toutes les distances ce qui constitue un gain non négligeable en temps de calcul.

Dans le cadre de ce travail, nous avons choisi de ne retenir que l'ACC qui nous a semblé adaptée à la visualisation de répartitions de points en grande dimension qui peuvent être issues soit d'une base de données soit de la simulation numérique. Nous pourrons alors visualiser ces points dans un espace de dimension inférieure sans grande perte d'information locale puisque le principe même de l'ACC est de conserver les distances les plus faibles. Ainsi, nous savons que deux points proches dans l'espace initial en D dimensions auront leurs distances conservées dans l'espace en p dimensions avec ($p < D$), ce qui nous permet de compléter les objectifs de l'ACC par la visualisation de deux ou plusieurs points très proches (amas). Si nous choisissons p égal à 2, nous disposerons d'un aperçu visuel de la répartition des points initiale dans l'espace des variables.

Dans un premier temps, nous appliquerons l'ACC sur des exemples en 3 dimensions, puis nous vérifierons que cette méthode peut être utilisée pour la visualisation des amas. Nous analyserons ensuite les différentes méthodes de visualisation puis nous comparerons une projection de points obtenues par ACC à celle résultant de l'ACP.

2.1.8.3 Illustrations de l'ACC

Parmi les nombreuses méthodes de visualisation de données, nous avons choisi de ne retenir que l'Analyse en Composantes Curvilignes qui semble répondre à nos problématiques. Après ce que nous avons observé sur l'exemple du swissroll qui montrait une bonne conservation des distances, nous nous sommes interrogés sur le réel bien fondé de cette méthode pour la visualisation de données en grande dimension. Pour valider ces conclusions, nous avons construit des exemples de complexité croissante mais pour en faciliter l'interprétation, nous avons volontairement choisi de travailler dans un espace initial à 3 dimensions.

Cas simulé N°1 : étude de deux spirales en trois dimensions

Dans cet exemple, l'espace d'entrée est en trois dimensions. Nous considérons un ensemble de points décrivant deux spirales planes dans le plan (x_1, x_3) plus ou moins espacées d'une distance d sur le troisième axe. La figure 1.15 représente la projection obtenue par ACC en faisant varier d.

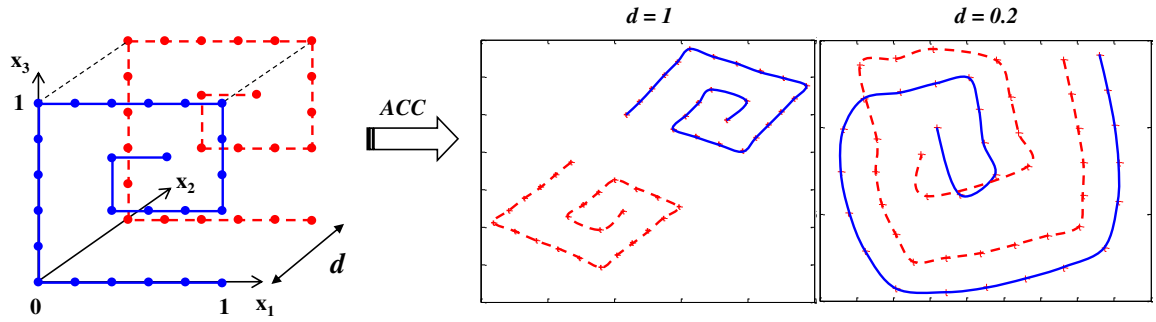


FIGURE 1.15 – Application de l'ACC sur un ensemble de points répartis sur deux plans plus ou moins espacés d'une distance d.

Sur la figure 1.15, nous observons que quelle que soit la distance d séparant les deux spirales, la forme est conservée dans l'espace de projection. Lorsque les spirales sont espacées d'une unité sur l'axe x_2 celles-ci sont bien séparées. Si nous rapprochons les spirales jusqu'à une distance égale à la distance séparant les points au sein d'une spirale ($d = 0.2$), les spirales projetées sont entremêlées car la distance entre les points d'une spirale est égale à la distance entre les deux spirales et l'ACC. Ce phénomène s'explique par l'objectif même de l'ACC qui cherche à conserver au mieux les distances minimales.

Cas simulé N°2 : étude de trois spirales en trois dimensions

Dans cet exemple, nous reprenons le cas présenté précédemment en ajoutant une spirale. Ainsi, nous considérons trois spirales planes dans le plan (x_1, x_3) plus ou moins espacées sur l'axe x_2 (figure 1.16). La spirale bleue est placée dans le plan $x_2 = 0$, la spirale violette dans le plan $x_2 = 0.2$ et la spirale rouge dans le plan $x_2 = 1$.

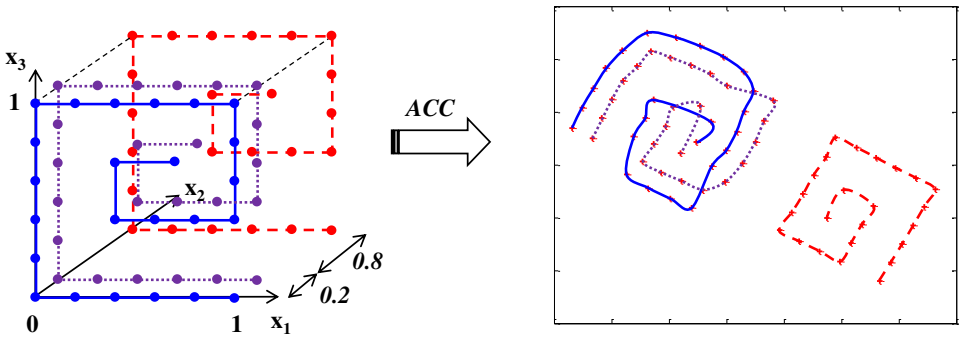


FIGURE 1.16 – Application de l'ACC sur un ensemble de points répartis sur trois plans plus ou moins espacés.

Nous constatons que les spirales les plus proches ($x_2 = 0, x_2 = 0.2$) sont entremêlées alors que la spirale placée dans le plan $x_2 = 1$ est isolée par l'algorithme. Ainsi, le fait d'ajouter des points (par une nouvelle spirale) ne perturbe pas l'ACC qui cherche toujours à projeter les données dans un espace de dimension inférieure en favorisant la proximité entre points.

Cas simulé N°3 : visualisation des amas

L'objectif de l'ACC étant de favoriser la conservation des faibles distances lors de la projection, nous nous sommes demandé si cette méthode ne pouvait pas être utilisée pour visualiser des amas de points. Pour répondre à cette question, nous avons considéré un ensemble de points distribués sur les faces d'un cube auquel nous avons ajouté des points extrêmement proches d'un sommet et du centre d'une face du cube afin de constituer des amas. La figure 1.17 représente cette distribution de points avec un premier amas sur un sommet du cube constitué de 7 points et un deuxième amas à 4 points situé au centre d'une face.

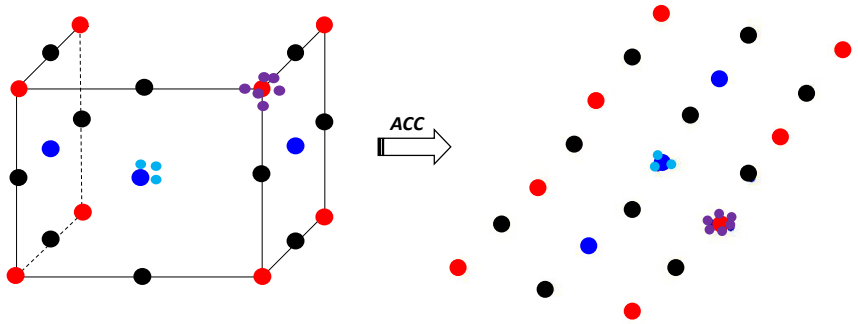


FIGURE 1.17 – Visualisation des amas par ACC sur un exemple en trois dimensions.

La projection de points obtenue par ACC permet non seulement de retrouver la proximité entre points mais aussi de visualiser les amas qui sont constitués de points que nous avons voulu extrêmement proches. Ainsi, la projection par ACC semble être une méthode efficace pour visualiser des points très proches dans une distribution.

Dans la suite de ce travail, nous proposerons d'autres exemples en grande dimension, issus de données réelles telles que des études QSAR ou issues de la simulation numérique pour des études en 20D et pouvant aller jusqu'à 50D. L'ACC sera alors utilisée pour visualiser les amas de points dans ces grands espaces.

2.2 Synthèse et analyse des méthodes de visualisation des données

Dans ce chapitre, nous avons présenté des méthodes qui permettent de visualiser des données en grande dimension par une réduction de la dimensionnalité. Ici nous nous sommes intéressés qu'aux méthodes non linéaires qui se différencient par le critère que l'on cherche à minimiser. Nous proposons maintenant de résumer les principes, les avantages et les inconvénients des différentes méthodes présentées.

- Les méthodes telles que MDS, les cartes de Sammon ou encore Isomap, projettent les données dans un espace de dimension inférieure en conservant au mieux les **propriétés globales** :
  - il existe deux variantes de la méthode **MDS** : la première est qualifiée de métrique et cherche à faire correspondre au mieux les distances de sortie aux distances d'entrée alors que la variante non métrique relie les distances par une fonction monotone f,
  - les **cartes de Sammon** minimisent un autre critère E mais ne permettent pas de bien conserver les faibles distances entre points,
  - la méthode **Isomap** propose de considérer des distances géodésiques qui nécessitent de définir le voisinage de chaque point. Pour cela nous devons définir soit le nombre de plus proches voisins (k), soit fixer une distance λ en deçà de laquelle les points sont considérés comme voisins.

- D'autres méthodes reposent sur la conservation des **propriétés locales** :
  - l'**ACC** permet de projeter les données en conservant les faibles distances et nécessite de fixer des paramètres : la fonction de coût F qui est directement liée à la distance critique λ en deçà de laquelle l'algorithme va privilégier la conservation des faibles distances et le facteur d'adaptation noté α. L'algorithme originel est "lourd" en termes de calculs mais cette limitation est résolue par l'utilisation de la quantification vectorielle qui permet de réduire le coût algorithmique,
  - la méthode **LLE** projette les données par reconstruction des points par leurs plus proches voisins. Les différentes étapes de cet algorithme demandent de minimiser deux fonctions : l'erreur de reconstruction d'un point et la fonction de coût notée $\Phi(Y)$. L'utilisation de cette méthode requiert de fixer le nombre de voisins k ou la distance λ.

- La méthode **RPM** se différencie des autres méthodes pré-citées car elle projette les données par l'intermédiaire d'un **système à particules**. Ceci implique de minimiser l'énergie potentielle qui dépend de la rigidité du système (notée p_r). Les données seront alors cartographiées sur une surface torique qui sera par la suite dépliée pour obtenir une carte en deux dimensions.

- Les méthodes **SOM** et **GTM** diffèrent des autres méthodes par leur objectif qui n'est pas de conserver les distances mais de fournir une information sur les relations de voisinage entre points. Ces cartes seront alors généralement utilisées pour la **classification des données**. Le principal inconvénient des cartes de Kohonen est d'imposer *a priori* la géométrie du réseau de neurones alors que les cartes GTM placent librement les neurones dans l'espace des variables.

L'ensemble de ces méthodes repose sur la minimisation d'un critère propre à chaque méthode mais peut nécessiter de définir des paramètres. Nous proposons de regrouper ces informations dans le tableau 1.1.

Tableau 1.1 – Tableau récapitulatif des méthodes de visualisation des données.

Méthodes	Paramètres à définir	Critères à minimiser
MDS métrique	aucun	$E_{MDS} = \sum_{i<j}(X_{ij} - Y_{ij})^2$
MDS non-métrique	fonction f	E_{MDS} avec $X_{ij} < X_{hk} \Rightarrow f(X_{ij}) < f(X_{hk})$
Cartes de Sammon	aucun	$E_{Sammon} = \frac{1}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} X_{ij}} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{(X_{ij}-Y_{ij})^2}{X_{ij}}$
Isomap	k ou λ	calcul du plus court chemin géodésique
ACC	F, λ, α	$E_{ACC} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(X_{ij} - Y_{ij})^2 F_\lambda(Y_{ij})$
LLE	k ou λ	$\Phi(Y) = \sum_i \left\|y_i - \sum_j W_{ij} y_j\right\|^2$
RPM	p_r	$E_{p_r} = \sum_{i<j} \frac{X_{ij}}{p_r Y_{ij}^{p_r}}$
SOM	Géométrie du réseau de neurones, Nombre de neurones, μ, f	
GTM	Nombre de neurones, μ, f	

Pour mieux comparer les réelles performances de chacune des méthodes, nous avons construit un cas d'étude constitué d'un ensemble de 21 points distribués sur les faces d'un cube et la figure 1.18 montre les différentes représentations.

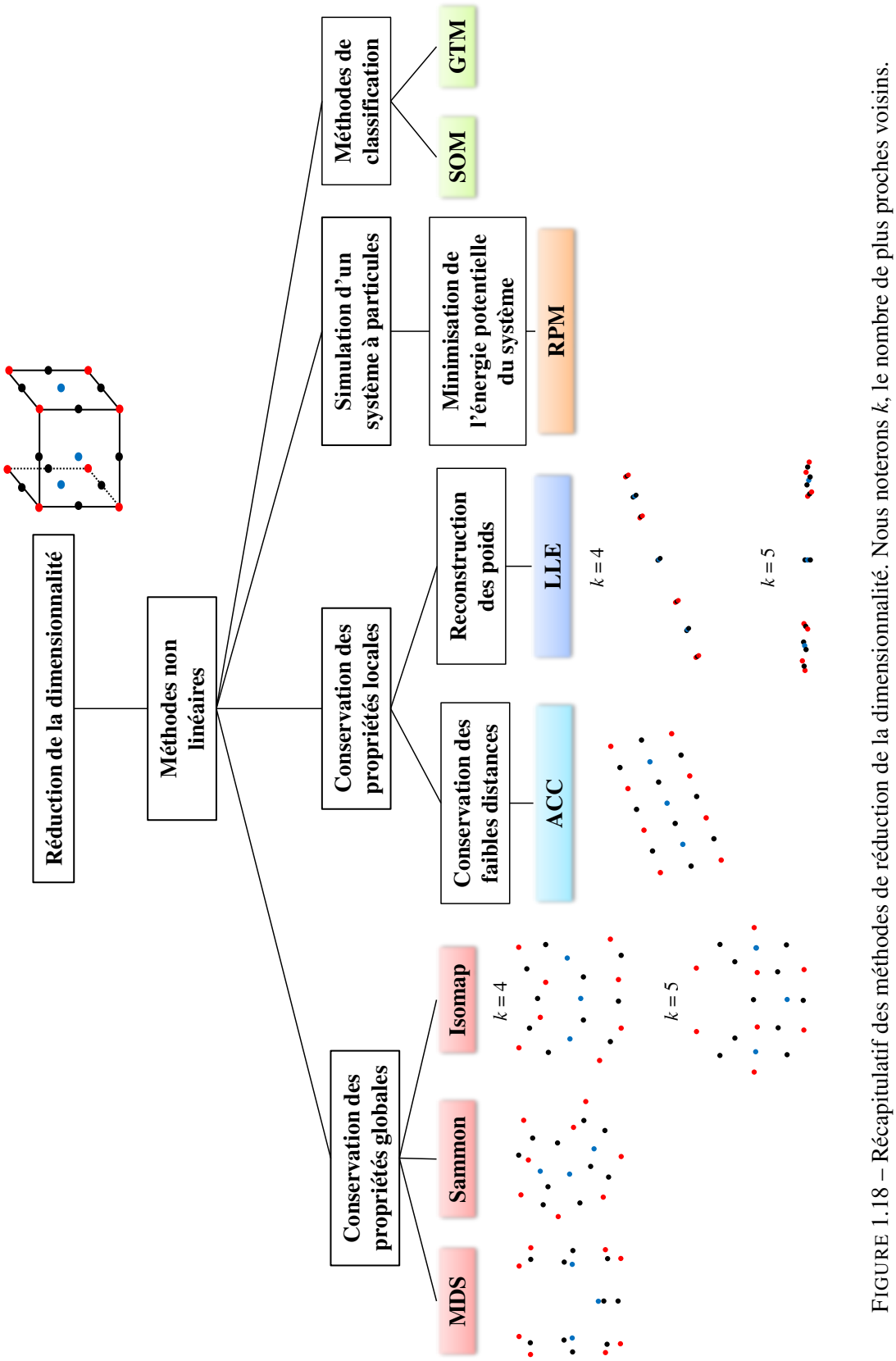


FIGURE 1.18 – Récapitulatif des méthodes de réduction de la dimensionnalité. Nous noterons k, le nombre de plus proches voisins.

Cette figure confirme que l'Analyse en Composantes Curvilignes (ACC) est la mieux adaptée à la visualisation des données. Pour compléter l'étude, nous avons comparé les résultats de l'ACC à ceux d'une méthode linéaire classiquement utilisée comme l'Analyse en Composantes Principales (ACP) [10]. Pour cela, nous les avons appliquées sur un jeu de données constitué de 41 points dans un espace en trois dimensions, distribués sur les faces d'un cube avec des points situés sur les sommets (en rouge), sur les arêtes (en noir) et au milieu des faces (en bleu). La figure 1.19 illustre cet ensemble de points et les projections obtenues par ACP et ACC.

Sur la figure 1.19, nous observons que les projections des 41 points par ACP et ACC sont différentes. En effet, lorsque nous utilisons l'ACP, nous retenons trois composantes principales, qui représentent respectivement 38.5%, 30.8% et 30.7% soit la totalité de l'inertie des données. Nous pouvons alors visualiser ces données sur trois graphiques. Dans le plan (CP1, CP2) et dans le plan (CP2,CP3) nous visualisons 17 points alors que dans le plan (CP1,CP3) nous n'en distinguons que 13. Ainsi, par ACP quel que soit le plan de projection, la superposition des données dans les plans de projection ne permet pas de visualiser l'ensemble des points candidats, ce qui implique une perte d'information quant à la répartition initiale des données.

A contrario, l'utilisation de l'ACC permet de visualiser les 41 points dans un seul plan de projection en fonction des composantes curvilignes avec la conservation des distances entre points, c'est-à-dire que les points qui étaient initialement proches restent proches dans la projection. Ainsi, à partir de cet exemple, nous pouvons confirmer que l'ACC permet de conserver la proximité entre les points.

FIGURE 1.19 – Comparaison ACP / ACC.

2.3 Les avancées des méthodes de visualisation

Nous avons choisi de nous intéresser particulièrement à l'Analyse en Composantes Curvilignes (ACC) pour visualiser des amas de points dans un espace multidimensionnel, mais cette simple visualisation peut s'avérer parfois insuffisante et il serait alors nécessaire de disposer d'outils quantitatifs renseignant sur la présence d'amas.

Pour cela, nous proposons de construire en parallèle une distribution de points en deux dimensions avec un nombre à évaluer et qui servira de "référence". Cette distribution de référence est obtenue par l'algorithme WSP (présenté page 62), qui est un algorithme de sélection visant à répartir des points uniformément dans un espace. Les critères intrinsèques d'uniformité de ce plan sont calculés pour déterminer les valeurs *Mindist* et *MoyMin* de cette distribution de référence.

Nous considèrerons comme critères d'évaluation d'une part le rapport R_{min} entre le minimum des distances minimales entre deux points (*Mindist*) de la distribution de points obtenue par projection ACC et celui de la distribution de référence (équation (2.13)) et d'autre part, le rapport R_{moy} entre la moyenne des distances minimales entre deux points (*MoyMin*) de la distribution de points obtenue par projection ACC et celle de la distribution de référence (équation (2.14)).

$$R_{min} = \frac{\textit{Mindist} \text{ de la distribution de points obtenue par ACC}}{\textit{Mindist} \text{ de la distribution de référence}} \tag{2.13}$$

$$R_{moy} = \frac{\textit{MoyMin} \text{ de la distribution de points obtenue par ACC}}{\textit{MoyMin} \text{ de la distribution de référence}} \tag{2.14}$$

Les valeurs de ces ratios nous donnent une indication sur la qualité de la répartition des points : si R_{min} est proche de 1 alors la distribution est proche de la référence, en revanche une valeur proche de 0 indique la présence de points très proches et donc d'amas. D'autre part, une valeur R_{moy} équivalente à celle du R_{min} indique une homogénéité de la répartition des distances minimales alors que la relation $R_{min} < R_{moy}$ caractérise la présence de points plus proches que ceux de la distribution de référence. En résumé, si les ratios sont proches de 1, nous pouvons considérer que la répartition des points est proche de la référence et donc de l'uniformité.

Afin d'illustrer l'utilisation de ces critères R_{min} et R_{moy}, nous considérons une distribution uniforme de 186 points dans un espace en 20 dimensions à laquelle nous avons délibérément ajouté 14 points supplémentaires proches des points déjà présents pour ainsi créer des amas et tester les performances de la méthode. Une simple représentation graphique pour visualiser la répartition des points et ainsi détecter d'éventuels amas n'est pas suffisante car nous ne pouvons envisager que des plans de coupe en 2 dimensions. Si nous choisissons d'appliquer l'ACC à la distribution initiale en 20 dimensions, nous obtenons une projection des points dans un espace à deux dimensions qui nous permet de visualiser rapidement la présence d'amas (figure 1.20).

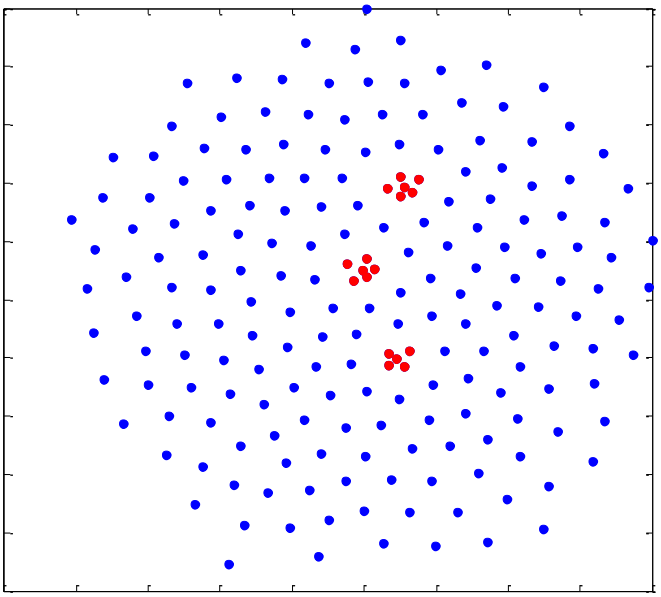


FIGURE 1.20 – Projection ACC d'une distribution de 200 points en 20 dimensions présentant des amas. Les points rouges représentent les points constituant les amas.

Après avoir projeté la distribution de points en grande dimension dans un espace de dimension inférieure par ACC, il est intéressant de calculer les critères R_{min} et R_{moy}. Dans cet exemple, les valeurs *Mindist* et *MoyMin* sont respectivement égales à 0.015 et 0.061 alors que celles de la distribution de référence (en 2D et 200 points) sont respectivement égales à 0.066 ce qui conduit à des valeurs $R_{min} = 0.22$ et $R_{moy} = 0.93$. La faible valeur R_{min} confirme la présence d'amas et la valeur élevée de R_{moy} indique que la distribution globale des points est proche de la référence. Ainsi, par ces critères nous sommes capables d'évaluer la qualité de cet ensemble de points et nous retrouvons parfaitement les indications de la méthode de construction de cette distribution à savoir l'ajout de points très proches à un ensemble de points répartis uniformément.

La difficulté rencontrée ici est le manque de précision du terme "amas". En effet, nous pouvons nous demander à partir de quelle distance des points peuvent être considérés comme suffisamment proches pour constituer un amas ou au contraire suffisamment éloignés pour ne plus être considérés comme un amas. Afin de préciser cette définition, nous nous sommes intéressés à l'évolution des critères d'uniformité lors de la suppression des amas. Pour cela, nous choisissons arbitrairement un point, puis nous éliminons tous les points situés à une distance inférieure à d de ce point, et nous répétons cette démarche sur les autres points tant qu'il est possible de supprimer des points. Cette opération est réitérée en faisant progressivement varier la distance d et nous obtenons ainsi différents sous-ensembles de points caractérisés par des valeurs de critères (*Mindist*, *MoyMin*, *Coverage*). La figure 1.21 représente l'évolution de ces critères en fonction de la distance d, et montre des paliers qui correspondent à des distributions plus uniformes. Par ailleurs, nous observons que pour de faibles valeurs de d, la valeur *Mindist* est inférieure à la valeur *MoyMin* ce qui confirme la présence d'amas jusqu'à l'obtention de la distribution contenant 186 points pour laquelle la valeur *Mindist* est proche de la valeur *MoyMin*.

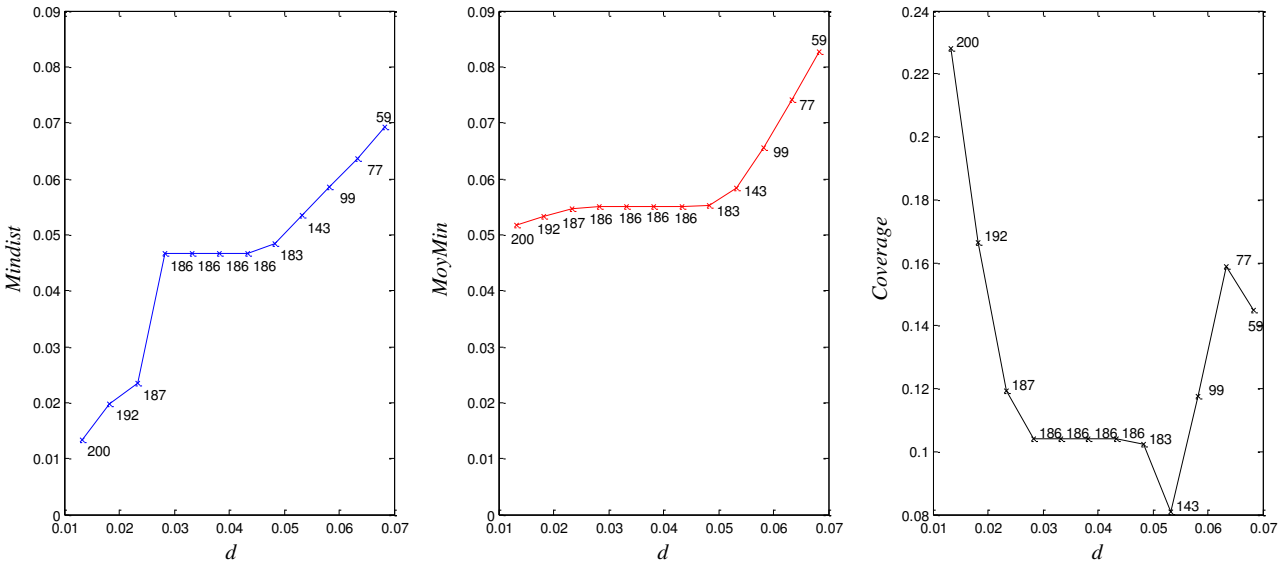


FIGURE 1.21 – Évolution des critères d'uniformité en fonction de la distance d. Les valeurs affichées sur la courbe correspondent au nombre de points restants.

D'après ces premiers résultats, nous pouvons proposer une distribution de meilleure qualité à 186 points qui correspond à la distribution initiale à laquelle nous avions délibérément ajouté des amas.

Dans la deuxième partie de ce manuscrit, nous proposerons d'utiliser l'ACC pour détecter et visualiser des amas sur des cas réels issus d'études QSAR, puis sur des plans uniformes construits en 20 dimensions ou dans une étude de "repliage".

2.4 Conclusion des méthodes de visualisation des données

A partir des méthodes de visualisation des données qui sont citées dans la littérature, nous avons pu établir une comparaison qui nous a menés à ne retenir que l'ACC. Contrairement aux autres méthodes, cette dernière présente l'avantage de projeter les données en fonction de leur proximité initiale. En effet, si deux points sont proches dans l'espace initial alors ils seront retrouvés proches dans l'espace de projection.

Cependant, lorsque nous souhaitons évaluer la qualité de la répartition des données initiale, la simple visualisation des données devient insuffisante et met en exergue le besoin d'outils complémentaires. Pour ce faire, nous proposons de calculer les ratios R_{min} et R_{moy} qui permettent de comparer les projections aux données d'une distribution de points uniforme. Nous avons montré que l'utilisation de ces critères en complément de la visualisation des données permet de confirmer la présence de points très proches constituant un amas, mais peut aussi signifier que les points peuvent être plus proches que ceux constituant la distribution uniforme. Par ailleurs, l'utilisation simultanée de ces deux critères apporte aussi une information quant à la distribution globale des points. Par exemple, un amas entrainera une faible valeur du ratio R_{min} (proche de 0) mais s'il est accompagné d'une valeur élevée du ratio R_{moy} (proche de 1) alors cela signifie que si nous venions à éliminer cet amas, la distribution de points est proche d'une distribution uniforme.

Par ces critères, nous sommes alors capables de compléter l'interprétation des données qui jusqu'ici ne pouvait être que visuelle en apportant de nouvelles informations quant à la distribution initiale des points à savoir la présence d'amas et le conditionnement global des points.

Chapitre 3

Les méthodes de sélection

Comme nous l'avons dit en introduction, dans de nombreux domaines d'application, il est courant de travailler avec des "tableaux" de grande dimension qui rendent l'interprétation des données difficile. Une solution envisageable pour simplifier l'analyse et l'interprétation de ces grands tableaux de données serait d'effectuer une sélection permettant de conserver l'information la plus explicative. Selon les données et l'objectif de l'étude, nous chercherons à réduire la dimension en sélectionnant soit des lignes, soit des colonnes.

Dans ce chapitre, nous présenterons tout d'abord quelques méthodes de sélection qui ne retiennent qu'un sous-ensemble de lignes (expériences ou simulations). L'objectif est alors d'extraire un sous-ensemble de points le plus représentatif possible de l'ensemble initial. Après une étude bibliographique des méthodes couramment utilisées, nous nous sommes intéressés aux cas particuliers des domaines tronqués et des sous-domaines d'un plus grand intérêt. Pour cela, nous aurons besoin d'algorithmes permettant de densifier la ou les zones d'intérêt.

Dans une deuxième partie, nous présenterons les méthodes de sélection de variables, permettant d'identifier puis d'éliminer les variables redondantes ou corrélées qui peuvent pénaliser les performances d'un modèle. Cette étape permet de simplifier un modèle ou un métamodèle et d'améliorer ses capacités prédictives .

3.1 Les méthodes de sélection de points

Les méthodes de sélection de points peuvent être utilisées dans deux cas de figure.

Tout d'abord, dans de nombreux domaines, le développement des appareils de mesure facilite l'acquisition de données ce qui entraîne une surabondance d'information et nécessite donc des outils performants pour extraire un sous-ensemble de résultats et en simplifier l'interprétation. L'objectif ici est donc de réaliser un échantillonnage approprié qui permettra, avec un nombre réduit de données, de conserver une information de bonne qualité c'est-à-dire la plus représentative possible de l'ensemble initial.

Dans le cadre de la modélisation, nous chercherons à diviser les données en deux sous-ensembles : un pour la calibration et un pour la validation du modèle. Le sous-ensemble de calibration doit être représentatif des données initiales et les points constituant la validation doivent être choisis pour évaluer la qualité du modèle. Dans ce contexte, la sélection aléatoire pour le sous-ensemble de calibration n'est pas suffisante car elle ne garantit ni sa représentativité de l'ensemble initial ni de lui inclure les points extrêmes dans l'espace des variables qui sont susceptibles de présenter des comportements particuliers qu'il est important de considérer dans l'étape d'apprentissage.

Il existe de nombreux algorithmes de sélection de points qui se différencient principalement par leur technique de base : certains reposent sur des calculs de distance alors que d'autres utilisent des clusters de points dans l'espace. Les méthodes basées sur les distances entre points sélectionnent un sous-ensemble de points dans un ensemble initial en considérant les distances entre les points. L'objectif des méthodes basées sur les clusters est de regrouper les données en clusters et à partir des résultats du "clustering", de choisir les objets représentatifs de l'ensemble initial pour chaque cluster. Tout d'abord, nous présenterons certaines de ces méthodes, les plus représentatives, avec leur algorithme respectif :

- Kennard et Stone,
- DUPLEX,
- DBOD,
- Optisim,
- WSP,
- DBSCAN,
- *k*-means.

Nous comparerons ensuite les performances de ces méthodes au regard des critères de qualité intrinsèques et résumerons leurs principaux avantages et inconvénients respectifs qui nous ont mené à proposer des améliorations à l'une de ces méthodes. Nous présenterons alors les nouvelles avancées que nous proposons pour pallier les inconvénients de ces algorithmes de sélection de points.

3.1.1 Méthodes de sélection de points basées sur les distances

Les méthodes de construction de plans uniformes reposant sur des critères de distance, considèrent généralement des distances euclidiennes.

3.1.1.1 Algorithme de Kennard et Stone (KS)

L'algorithme de Kennard et Stone [28] est une méthode séquentielle qui permet d'extraire un sous-ensemble de N points d'un ensemble de N_c points candidats en D dimensions. A chaque itération, l'algorithme sélectionne le point le plus éloigné des points déjà retenus. L'algorithme peut être décrit ainsi (Algorithme 3.1) :

Algorithme 3.1 Algorithme de Kennard et Stone

Considérer un ensemble de N_c points candidats dans l'espace à D dimensions
Calculer la matrice des distances euclidiennes de l'ensemble des points candidats :

$$d_{ij} = \|x_i - x_j\| = \sqrt{\sum_{r=1}^{D} (x_{ir} - x_{jr})^2} = \text{distance euclidienne entre les points } i \text{ et } j$$

Choisir les points I et J tels que : $d_{IJ} = \max(d_{ij})$

JUSQU'A ce que $N = N_c$ où N = nombre de points souhaités

- Calculer les distances des$(N_c - N)$ points restants par rapport aux N points choisis et retenir la valeur minimale :

  $$\Delta_i(N) = \min\{d_{i1}, d_{i2}, ..., d_{iN}\}$$

  avec $\Delta_i(N)$, la distance du point candidat i non encore dans la matrice des points sélectionnés au point le plus proche dans les points sélectionnés.
- Pour le $(N+1)^{ème}$ point de la matrice, choisir parmi les $(N_c - N)$ points candidats restants celui pour lequel :

  $$\Delta(N+1) = \max\{\Delta_i(N)\}$$

  pour que le nouveau point soit le plus éloigné des points déjà sélectionnés.

FIN

Nous pouvons facilement représenter la progression de l'algorithme de Kennard et Stone dans un espace à $D = 2$ dimensions ($N_c = 100$ points candidats) (figure 1.22) :

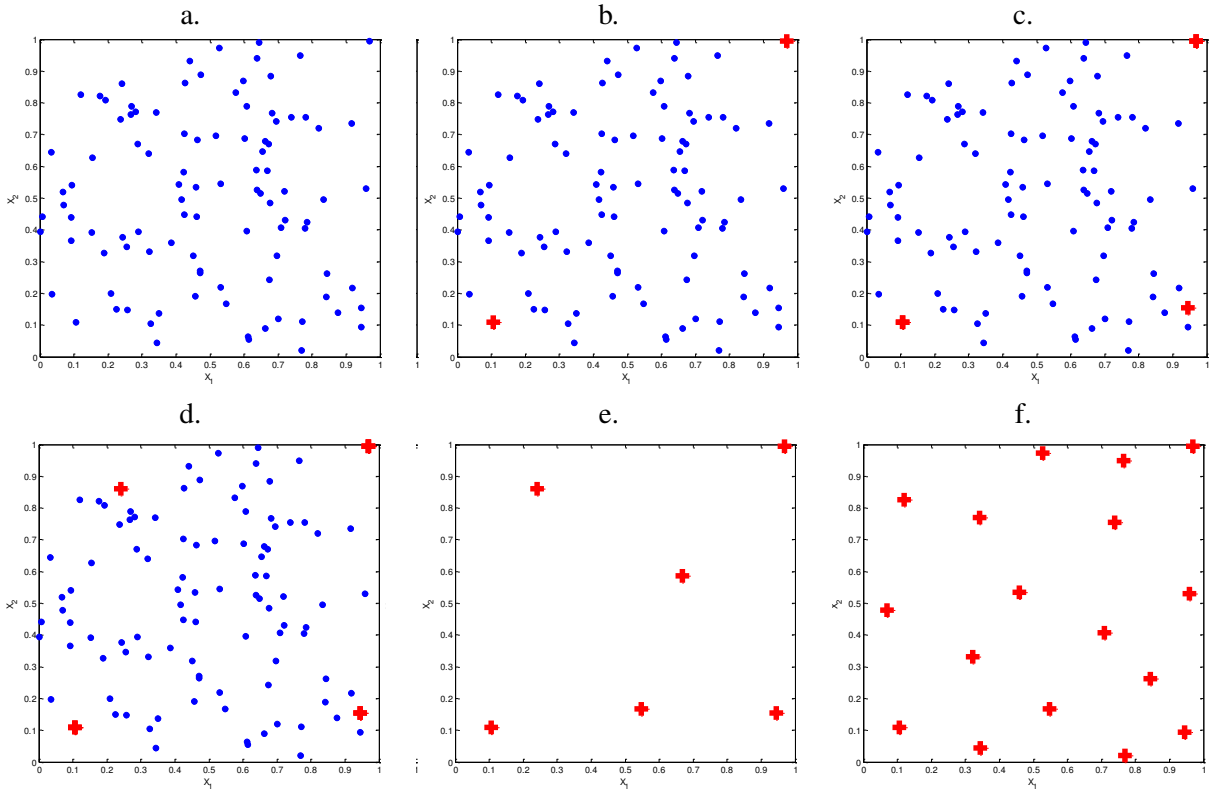


FIGURE 1.22 – Progression de l'algorithme de **Kennard et Stone**. **a)** Matrice candidate aléatoire en 2D et $N_c = 100$ points. **b)** Sélection des deux points les plus éloignés de l'ensemble des points candidats. **c)** Le troisième point sélectionné sera le point le plus éloigné des deux précédents. A chaque itération, l'algorithme sélectionne le point le plus éloigné des points déjà retenus jusqu'à sélectionner les N points désirés. **d)** Solution à $N = 4$ points. **e)** Solution à $N = 6$ points. **f)** Solution à $N = 17$ points.

Nous pouvons constater que pour une valeur de N donnée, la répartition des points dans l'espace n'est pas obligatoirement uniforme. Par exemple, pour $N = 6$, l'ensemble des points sélectionnés ne présente pas une répartition uniforme contrairement à la structure pour $N = 17$ points.

Il existe une variante de l'algorithme de Kennard et Stone [29] qui se différencie par la première étape. En effet, l'algorithme débute par la recherche du point candidat le plus proche du centre du domaine puis de son point le plus éloigné. A partir de ces deux premiers points, la procédure "classique" de l'algorithme de KS reprend c'est à dire par le troisième point sera le point le plus éloigné du centre de gravité des deux points précédemment sélectionnés. La progression de cette variante de l'algorithme de KS est représentée sur la figure 1.23.

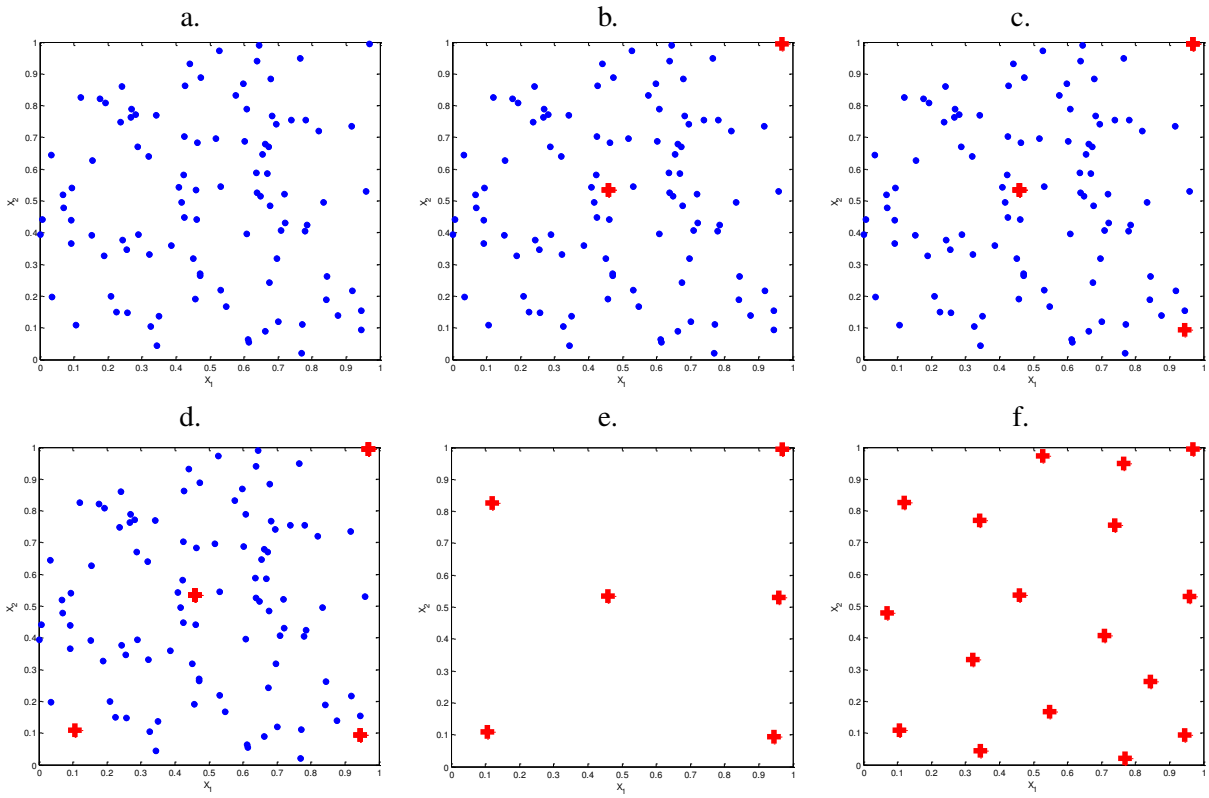


FIGURE 1.23 – Progression de l'algorithme de **Kennard et Stone** avec départ du point au centre. **a)** Matrice candidate aléatoire en 2D et $N_c = 100$ points. **b)** Sélection du point le plus proche du centre du domaine et du point le plus éloigné du centre de l'ensemble des points candidats. **c)** Le troisième point sélectionné sera le point le plus éloigné du centre de gravité des deux précédents. A chaque itération, l'algorithme sélectionne le point le plus éloigné des points déjà retenus jusqu'à sélectionner les N points désirés. **d)** Solution à $N = 4$ points. **e)** Solution à $N = 6$ points. **f)** Solution à $N = 17$ points.

Tout comme avec l'algorithme de KS classique, nous constatons que pour une valeur de N donnée, la répartition des points dans l'espace n'est pas obligatoirement uniforme. Par ailleurs, les distributions obtenues par les deux algorithmes de KS ne conduisent pas toujours aux mêmes solutions.

3.1.1.2 Algorithme DUPLEX

Snee [30] propose l'algorithme DUPLEX qui est une modification de l'algorithme de Kennard et Stone. DUPLEX construit en parallèle deux sous-ensembles de points, sélectionnés par l'algorithme de Kennard et Stone (Algorithme 3.2). Cet algorithme est souvent utilisé en spectroscopie pour construire les ensembles de calibration et de validation utilisés dans la modélisation des données. Cet algorithme, décrit ci-dessous, débute par la sélection des deux points les plus éloignés dans l'ensemble des points candidats, qui seront assignés à l'ensemble de calibration, puis dans les points restants, les deux points les plus éloignés seront attribués à l'ensemble de validation. L'alternance entre l'ensemble de calibration et l'ensemble de validation est poursuivie jusqu'à ce que tous les points candidats soient assignés à l'un des deux sous-ensembles.

Algorithme 3.2 Algorithme DUPLEX

Considérer un ensemble de N_c points candidats dans l'espace à D dimensions
Calculer la matrice des distances euclidiennes de l'ensemble des points candidats

$$d_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{r=1}^{D} (x_{ir} - x_{jr})^2}$$

Choisir les points I et J tels que : $d_{IJ} = \max(d_{ij})$, qui appartiendront au sous-ensemble de calibration.
Parmi les points candidats restants, choisir les points K et L tels que : $d_{KL} = \max(d)$. Les points K et L seront affectés au sous-ensemble de validation.

JUSQU'À ce que $N = N_c$

- Calculer les distances des $(N_c - N)$ points restants par rapport aux N points choisis du sous-ensemble de calibration :

$$\Delta_i(N) = \min \{d_{i1}, d_{i2}, ..., d_{iN}\}$$

- Pour le $(N+1)^{ème}$ point du sous-ensemble de calibration, choisir parmi les $(N_c - N)$ points candidats restants celui pour lequel :

$$\Delta(N+1) = \max \{\Delta_i(N)\}$$

- Calculer les distances des $(N_c - N)$ points restants par rapport aux N points choisis du sous-ensemble de validation :

$$\Delta_i(N) = \min \{d_{i1}, d_{i2}, ..., d_{iN}\}$$

- Pour le $(N+1)^{ème}$ point du sous-ensemble de calibration, choisir parmi les $(N_c - N)$ points candidats restants celui pour lequel :

$$\Delta(N+1) = \max \{\Delta_i(N)\}$$

FIN

La progression de l'algorithme DUPLEX dans un espace à $D = 2$ dimensions ($N_c = 100$ points candidats) peut facilement être représentée (figure 1.24) :

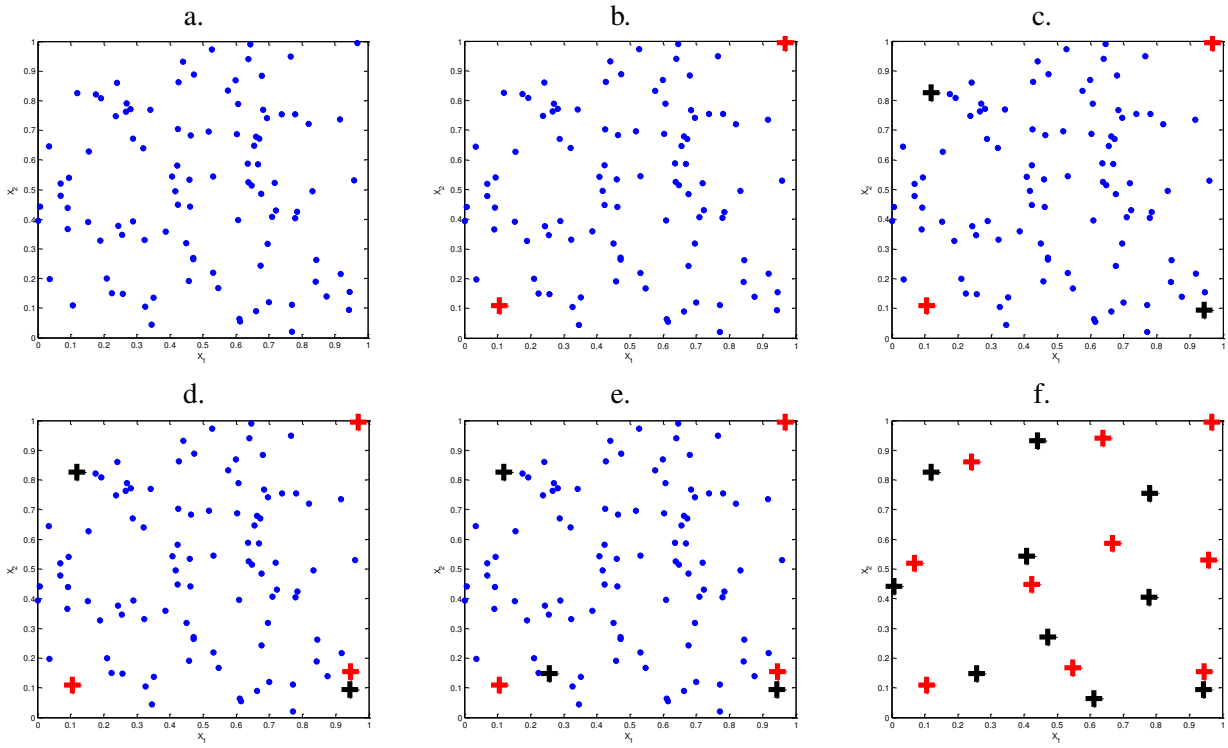


FIGURE 1.24 – Progression de l'algorithme **DUPLEX**. **a)** Matrice candidate aléatoire en 2D et $N_c = 100$ points. **b)** Sélection des deux points les plus éloignés de l'ensemble des points candidats qui seront affectés à l'ensemble de calibration. **c)** Sélection des deux points les plus éloignés parmi les points candidats restants qui seront affectés à l'ensemble de validation. **d)** Sélection du point le plus éloigné des deux premiers points sélectionnés pour l'ensemble de calibration. **e)** Sélection du point le plus éloigné des deux premiers points sélectionnés pour l'ensemble de validation. L'algorithme est poursuivi jusqu'à atteindre N le nombre de points désirés dans les sous-ensembles de calibration et de validation. **f)** Solution avec $N = 10$ points dans chaque sous-ensemble.

La méthode DUPLEX permet de construire en parallèle les sous-ensembles de calibration et de validation qui vont couvrir le domaine mais nous n'avons aucune information quant à la bonne répartition des points dans l'espace des variables et le principe de la méthode peut laisser supposer que les points des deux sous-ensembles seront proches.

3.1.1.3 Algorithme DBOD (Distance-Based Optimal Design)

Marengo et Todeschini [31] proposent une méthode pour la construction de plans d'expériences uniformes basée sur la maximisation de la distance minimale entre tous les points sélectionnés, par substitution séquentielle.

L'algorithme DBOD peut se résumer ainsi (Algorithme 3.3) :

Algorithme 3.3 Algorithme DBOD

Considérer un ensemble de N_c points candidats dans l'espace à D dimensions
Sélectionner arbitrairement un ensemble E_N contenant N points, avec N le nombre de points désiré

JUSQU'A ce que $dmax_i < d_i^*$

- Calculer les distances entre les N points appartenant à E_N
- Retenir d_i^* la distance minimale entre le point i appartenant à E_N et tous les autres points de E_N.

  - POUR chaque point j appartenant aux $(N_c - N)$ points candidats restants
    - Substituer le point i par le point j et créer un sous-ensemble temporaire T
    - Retenir la distance minimale d_j^* entre tous les points appartenant à T
  - FIN

- Calculer $dmax_i$ correspondant à la plus grande distance de l'ensemble des distances minimales obtenues par substitution du point i avec les $(N_c - N)$ points

$$dmax_i = max\left\{d_j^*\right\}$$

- SI
  - $dmax_i > d_i^*$, alors la substitution est favorable, donc réalisée, car la distance minimale augmente.
  - $dmax_i = d_i^*$, alors la substitution n'apporte pas de changement notable. Dans ce cas, la décision sera prise après calcul de la somme des distances entre les points du plan, avant et après la substitution. Si la substitution conduit à une augmentation de cette somme alors le nouveau point ajouté par substitution est globalement plus éloigné des autres et la substitution est effectuée.
  - $dmax_i < d_i^*$, alors la substitution est refusée car elle conduit à un plus mauvais conditionnement.
- FIN

FIN

Il est possible que plusieurs substitutions conduisent à la situation $dmax_i = d_i^*$. Dans ce cas, le test de la somme des distances est répété pour toutes les substitutions jusqu'à en obtenir une qui soit favorable. Si aucune n'est favorable, alors l'algorithme s'arrête.

L'algorithme DBOD repose sur des substitutions séquentielles à partir d'un sous-ensemble E_N choisi arbitrairement, il est donc difficile de détailler le chemin suivi par cette méthode de sélection. Toutefois, nous pouvons représenter les sous-ensembles à N = 10 et 17 points obtenus par la méthode DBOD (figure 1.25). Cet algorithme débutant par une sélection arbitraire du sous-ensemble E_N, nous proposons de comparer les sous-ensembles obtenus par deux exécutions distinctes de l'algorithme DBOD pour illustrer l'aspect stochastique de la méthode.

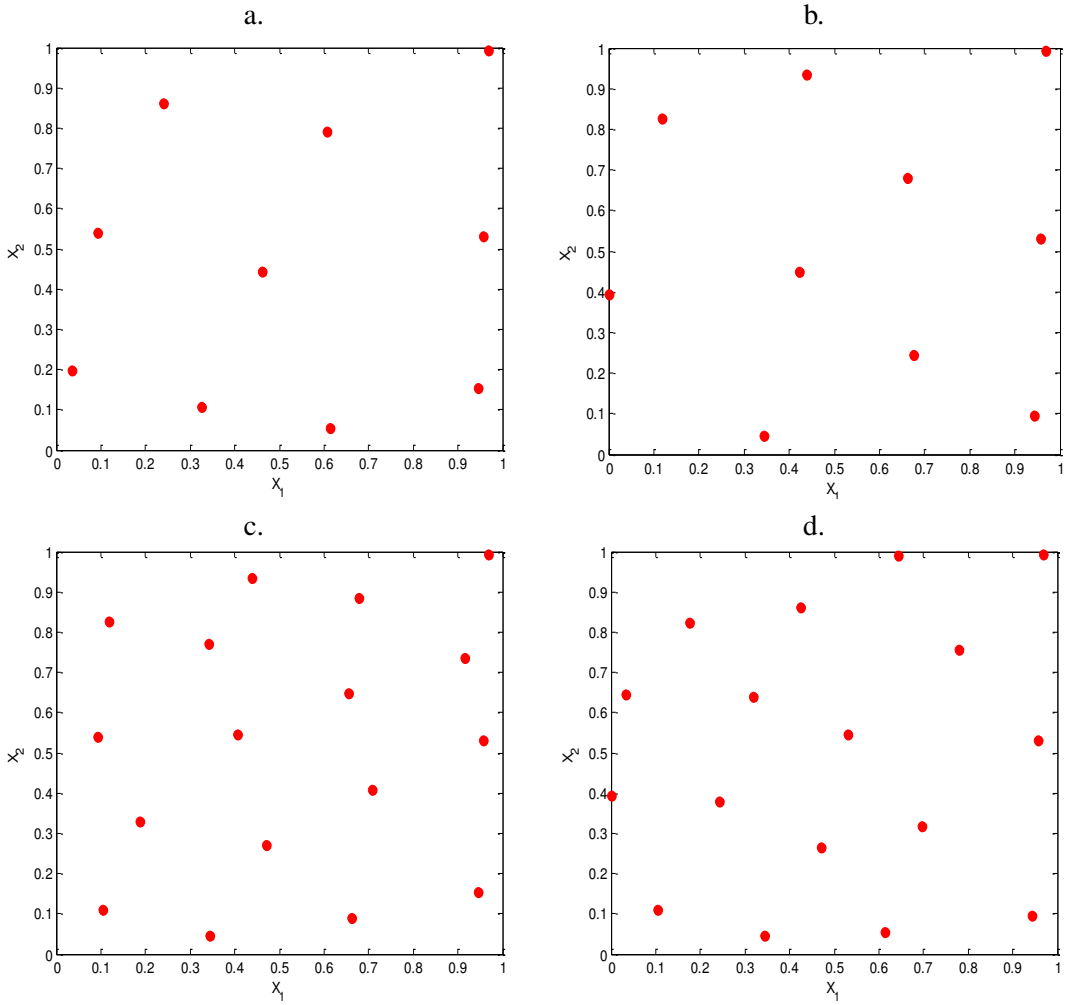


FIGURE 1.25 – Représentation des sous-ensembles obtenus par l'algorithme **DBOD** exécuté à partir de deux sous-ensembles E_N différents, pour $N = 10$ points (**a**) et **b**)) et pour $N = 17$ points (**c**) et **d**)).

La figure 1.25 montre que les sous-ensembles obtenus sont différents lorsque l'algorithme DBOD est appliqué sur la même matrice candidate et pour un même nombre N de points souhaité, ce qui s'explique par le choix arbitraire de l'ensemble E_N.

3.1.1.4 Algorithme OptiSim

La méthode de sélection OptiSim, proposée par Clark [32], repose sur une procédure de recyclage sur un ensemble de points choisis aléatoirement. Elle nécessite de définir au départ, le nombre N de points à sélectionner, la distance minimale ε entre deux points et S le nombre de points aléatoirement sélectionnés. Généralement, S est de l'ordre de 5% à 25% du nombre de points initial [29]. Quatre ensembles différents sont utilisés dans l'algorithme : la matrice X des N_c points candidats, la matrice X_N des N points sélectionnés, la matrice X_S des S points sélectionnés aléatoirement et X_r la matrice de recyclage.

L'algorithme OptiSim peut se résumer ainsi (Algorithme 3.4) :

Algorithme 3.4 Algorithme OptiSim

Définir les valeurs des paramètres : ε, N et S.
Choisir le point initial O qui sera le point le plus proche du centre de gravité de l'ensemble des N_c points candidats.
Supprimer le point O de la matrice X.

JUSQU'À obtenir N points dans X_N

- JUSQU'À obtenir S points dans X_S ou JUSQU' À ce qu'il n'y ait plus de points candidats
  - Choisir aléatoirement un point e_i parmi les points candidats de la matrice X.
  - Calculer la distance d entre e_i et le point O.
    - SI
      - $d \geq \varepsilon$, placer le point e_i dans la matrice X_S.
    - SINON
      - placer le point e_i dans la matrice X_r.
    - FIN
- FIN

- Placer dans X_N le point de X_S associé à la plus grande distance d.
- Tous les points restants dans X_S sont alors transférés vers X_r.
  - SI
    - X_N ne contient pas N points et que la matrice candidate X est vide, les points de X_r sont transférés vers X
  - FIN

FIN

Daszykowski et al. [29] proposent une valeur par défaut de ε qui est obtenue en considérant la fraction de points sélectionnés (N/N_c) et le volume V de l'hypersphère en D dimensions formée par le même nombre de points (N_c) que l'ensemble candidat mais uniformément distribués dans l'espace des variables. La figure 1.26 compare les sous-ensembles obtenus pour un même nombre de points N, une valeur de ε par défaut et une valeur de S égal à 5% ou 25% des données initiales. L'algorithme OptiSim repose sur un choix aléatoire des points à placer initialement dans X_s ou X_r, ce qui conduit à des solutions différentes pour des valeurs de paramètres identiques.

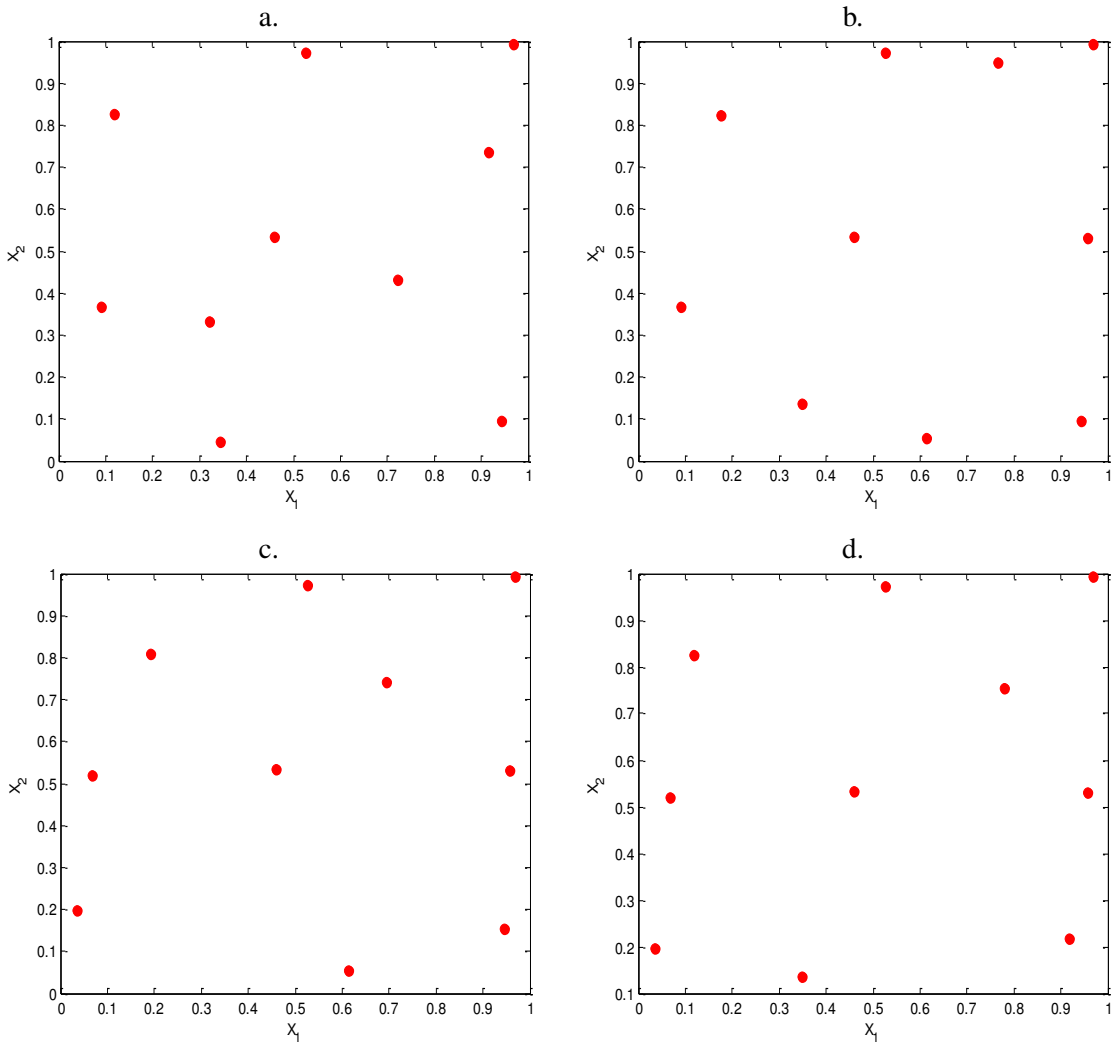


FIGURE 1.26 – Représentation des sous-ensembles à $N = 10$ points obtenus par l'algorithme **OptiSim** à partir d'une matrice candidate aléatoire en 2D et 100 points. Les sous-ensembles **a)** et **b)** sont obtenus pour $S = 5$ points (5%) et ε par défaut. Les sous-ensembles **c)** et **d)** sont obtenus pour $S = 25$ points (25%) et ε par défaut.

D'après la figure 1.26, nous constatons que les sous-ensembles obtenus par l'algorithme OptiSim pour une valeur de N fixée, peuvent présenter des répartitions de points différentes pour des valeurs de paramètres identiques.

3.1.1.5 Algorithme WSP

L'algorithme de Wootton, Sergent, Phan-Tan-Luu (WSP) [33, 34, 35, 36, 37] est un algorithme qui permet de sélectionner N points dans un ensemble initial de N_c points candidats, de telle manière qu'ils soient au moins distants d'une valeur d_{min}, choisie au préalable, des autres points déjà inclus dans le plan.

L'algorithme WSP peut se résumer ainsi (Algorithme 3.5) :

Algorithme 3.5 Algorithme WSP

Considérer un ensemble de N_c points candidats dans l'espace en D dimensions

Calculer la matrice des distances euclidiennes de l'ensemble des N_c points candidats

$$d_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{r=1}^{D} (x_{ir} - x_{jr})^2} = \text{distance entre les points } i \text{ et } j$$

Choisir un point initial O dans l'ensemble des points candidats

Fixer une valeur d_{min}

JUSQU'À ce que tous les points candidats soient sélectionnés ou éliminés

- Éliminer de l'ensemble candidat tous les points I tels que $d_{OI} < d_{min}$
- Placer le point O dans le sous-ensemble final
- Éliminer le point O de l'ensemble des points candidats et le remplacer par le point le plus proche parmi les points restants

FIN

Le nombre de points sélectionnés par l'algorithme WSP est directement lié à la valeur d_{min}. En effet plus la valeur d_{min} est petite, plus le nombre de points sélectionnés est important et inversement. Si le nombre de points à sélectionner est fixer au préalable, la valeur d_{min} sera ajustée par itération jusqu'à obtenir ou se rapprocher du nombre de points N désiré. La principale caractéristique de l'algorithme WSP est de fixer une distance minimale séparant tous les points sélectionnés, ce qui garantit une répartition uniforme du sous-ensemble final (avec de bons critères de qualité tels que *Mindist* et *Coverage*) quelle que soit la valeur de N. La progression de l'algorithme dans un espace à $D = 2$ dimensions ($N_c = 100$ points candidats) est représentée ci-dessous (figure 1.27).

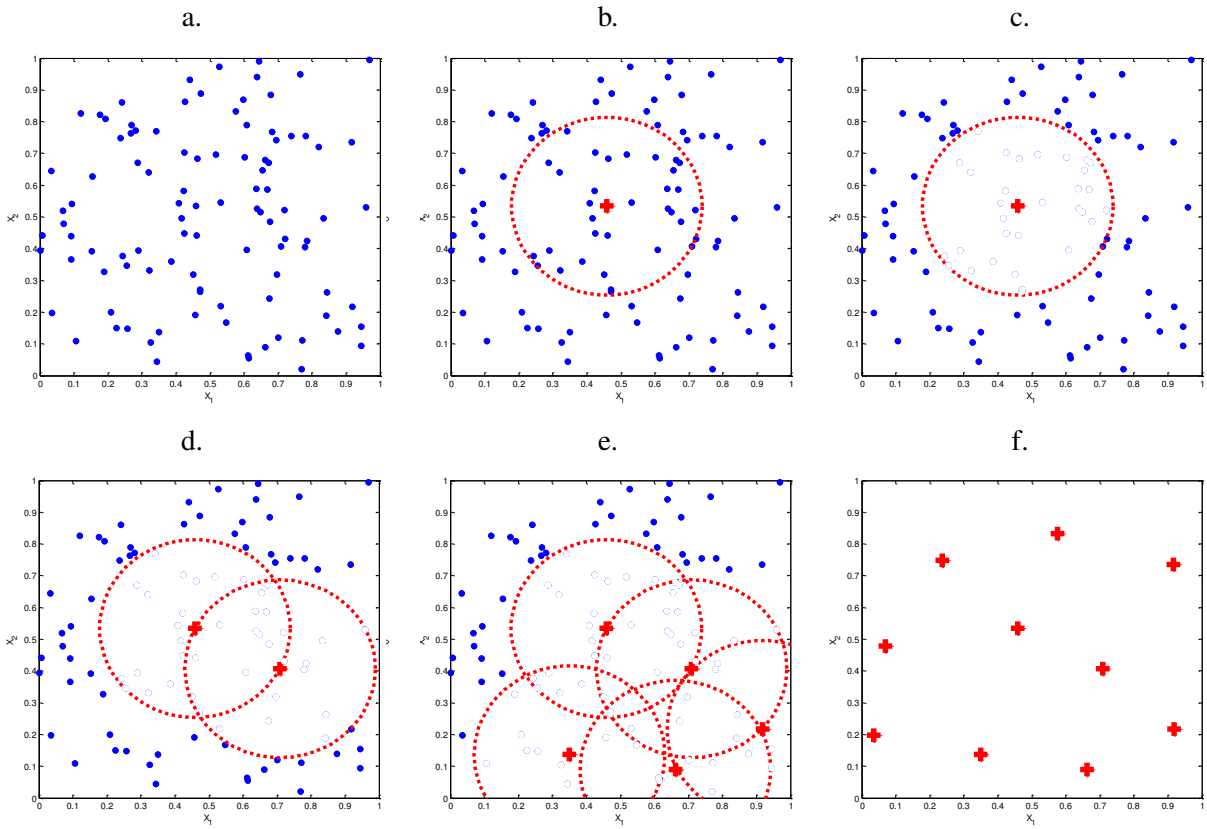


FIGURE 1.27 – Progression de l'**algorithme WSP**. **a)** Matrice candidate aléatoire en 2D et $N_c = 100$ points. **b)** Sélection du point initial qui est choisi ici comme le point le plus proche du centre du domaine. **c)** Élimination de tous les points situés à une distance inférieure à la valeur d_{min} puis le point le plus proche parmi les points restants du point précédemment retenu est sélectionné. L'algorithme est répété (**d)** et **e**)) jusqu'à ce qu'il n'y ait plus de points à éliminer. **f)** Solution à $N = 10$ points obtenue pour une valeur d_{min}= 0.28.

L'utilisation de l'algorithme WSP garantit l'obtention de solutions couvrant uniformément le domaine expérimental. A partir du même exemple, si nous diminuons la valeur d_{min} à 0.1 le nombre de points retenus par l'algorithme augmente à 40 points (figure 1.28).

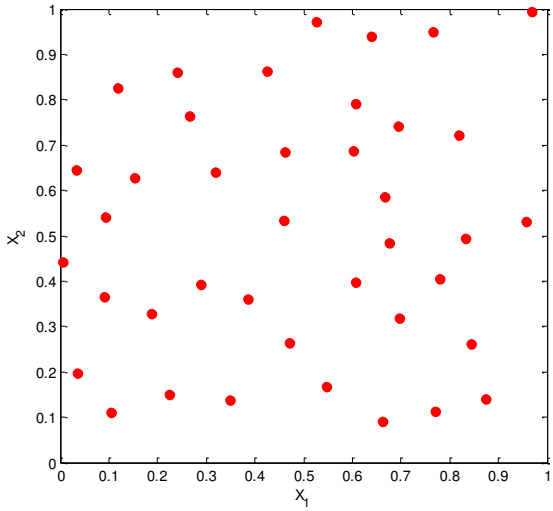


FIGURE 1.28 – Solution à N = 40 points obtenue par l'algorithme WSP pour d_{min} = 0.1.

Lorsque l'algorithme WSP est répété, les sous-ensembles obtenus sont identiques pour une valeur d_{min} fixée et pour le même point initial. Dans les exemples proposés ci-dessus, nous avons choisi le point le plus proche du centre du domaine comme point initial mais il est possible de choisir un autre point. La figure 1.29 propose les solutions obtenues pour une valeur d_{min} fixée en considérant différents points initiaux.

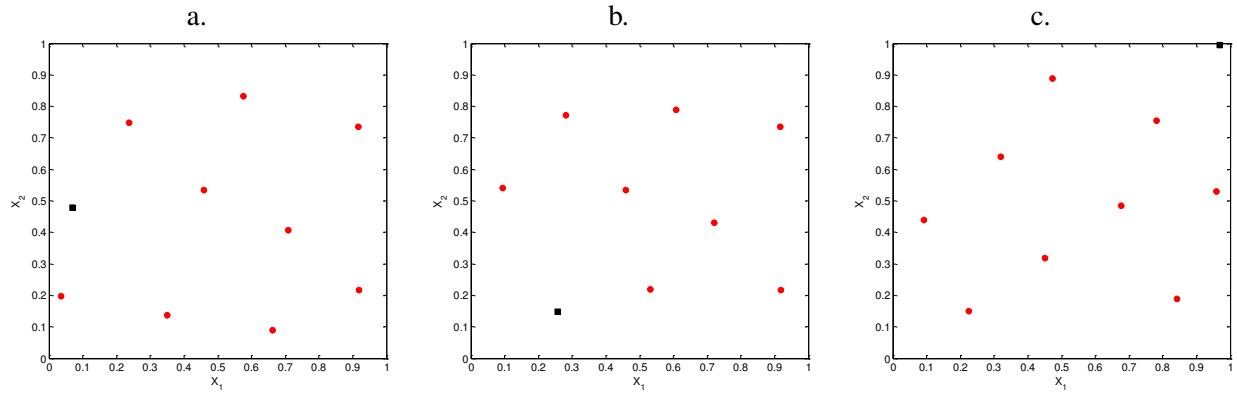


FIGURE 1.29 – Solutions obtenues par l'algorithme WSP en considérant différents points initiaux (représentés par un rectangle noir) et une valeur d_{min} fixée. **a)** Le point initial est le point au centre du domaine ou son plus proche voisin. **b)** Le point initial est un point choisi aléatoirement. **c)** Le point initial est le point le plus éloigné du centre du domaine.

Généralement le point initial O choisi est le point le plus proche du centre du domaine.Toutefois, lorsque la matrice candidate contient un nombre suffisant de points, quel que soit le point initial les résultats sont identiques. Par ailleurs, il a été montré que la qualité intrinsèque du sous-ensemble final ne dépend pas de la matrice candidate si et seulement si le nombre de points N_c est suffisant. Le nombre de points candidats N_c dépend du nombre de points dans le sous-ensemble final mais Santiago et al. [36] conseillent de choisir N_c qui doit au moins être égal à 5 à 10 fois le nombre de points désiré N dans le sous-ensemble final.

3.1.1.6 Méthode de Puchwein

L'algorithme proposé par Puchwein [38] sélectionne des points à partir d'un ensemble de N_c points candidats en supprimant de manière itérative les points similaires par l'utilisation des distances de Mahalanobis [39, 40]. L'algorithme débute par une Analyse en Composantes Principales (ACP) [10] qui aboutit à une nouvelle matrice candidate X_{ACP} de dimension $D_{ACP} < D$ et dont les éléments sont les scores de l'ACP. La méthode ensuite ne considère que cette matrice et l'algorithme de Puchwein utilise les distances de Mahalanobis. Afin de calculer les distances de Mahalanobis, il est nécessaire dans un premier temps de construire la matrice de covariance notée S (équation (3.1)). Pour cela, on considère une matrice X constituée de N_c points et D variables. Ce calcul nécessite de centrer au préalable chaque variable X_i sur la valeur moyenne $\bar{X}$ de la variable X_i considérée.

$$S = \frac{1}{N_c - 1}(X_i - \bar{X})^t (X_i - \bar{X}) \tag{3.1}$$

avec $i = 1, ..., D$

De manière générale, la distance de Mahalanobis est calculée entre un vecteur de données noté x_i et la valeur moyenne du nuage de données notée $\bar{x}$ qui peut être assimilée au barycentre des données (équation (3.2)).

$$d_M(x_i) = \sqrt{(x_i - \bar{x})^t S^{-1} (x_i - \bar{x})} \tag{3.2}$$

La première étape consiste à sélectionner dans X_{ACP} le point le plus éloigné du centre de gravité des données initiales par le calcul des distances de Mahalanobis. On choisit alors une distance limite en deçà de laquelle les points seront supprimés de l'ensemble des points candidats. Le point avec la plus grande distance de Mahalanobis est sélectionné et la procédure est répétée jusqu'à ce qu'il n'y ait plus de points à supprimer. Le nombre de points sélectionnés dépend alors de la valeur de la distance limite. Plus cette distance est faible, plus le nombre de points sélectionnés est grand et inversement. Cette procédure doit donc être réitérée pour différentes valeurs de la distance limite jusqu'à ce que le nombre de points sélectionnés atteigne ou se rapproche de la valeur souhaitée (Algorithme 3.6).

Algorithme 3.6 Méthode de Puchwein

Sélectionner dans X_{ACP} le point le plus éloigné du centre de gravité des données initiales par le calcul des distances de Mahalanobis.
Choisir une distance limite initiale d_{lim}

JUSQU'A ce que tous les points candidats soient sélectionnés ou éliminés
- SI
  - $d_M(x_i) < d_{lim}$, le point i est supprimé,
- SINON
  - le point i est sélectionné
- FIN

FIN

3.1.2 Méthodes de sélection de points basées sur les clusters

3.1.2.1 DBSCAN (Density Based Spatial Clustering of Applications with Noise)

La méthode DBSCAN [29, 41, 42] repose sur la densité des données et classe les points en trois catégories : les noyaux, les "objets limites" et les "outliers". Afin de déterminer à quelle catégorie appartient chaque point, il est nécessaire de définir deux paramètres : v le nombre de points dans le voisinage et ε le rayon du voisinage. Un point sera considéré comme un noyau (figure 1.30 a.) s'il a au moins v points dans son voisinage c'est-à-dire v points situés à une distance inférieure à ε. Un point sera considéré comme un "objet limite" (figure 1.30 b.) si dans son voisinage on trouve un noyau et moins de v points. Un point sera qualifié de "outlier" (figure 1.30 c.) si dans son voisinage il a moins de v points et pas de noyau : ces points sont considérés comme non significatifs et peuvent être assimilés à du bruit.

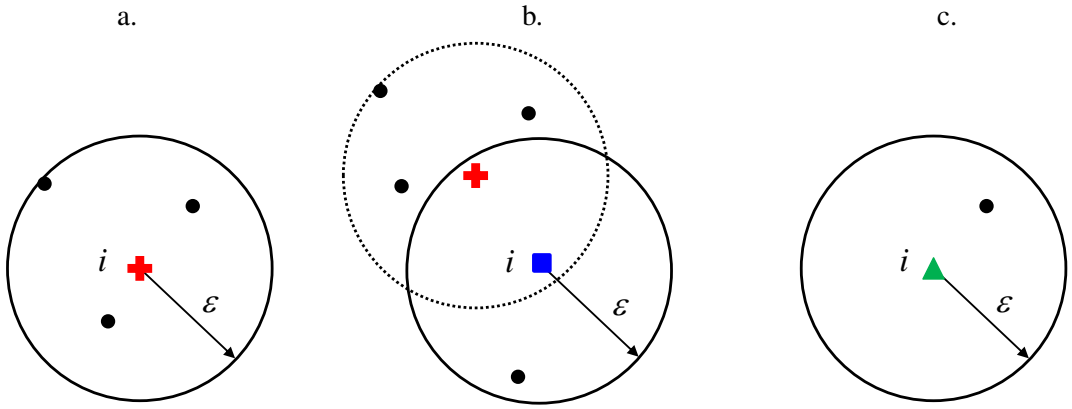


FIGURE 1.30 – Types des points en fonction de la densité de points dans le voisinage. Pour $v = 3$, le point i est : **a)** un noyau (croix rouge), **b)** un objet limite (rectangle bleu), **c)** un outlier (triangle vert).

L'algorithme DBSCAN peut se résumer ainsi :

- Le premier point est choisi aléatoirement
- Si ce premier point est un noyau alors il forme le cluster A et tous les points situés dans son voisinage c'est-à-dire à une distance inférieure à ε sont rattachés au cluster A. Ces points sont alors considérés comme "traités" par l'algorithme et ne pourront plus être affectés à un autre cluster.
- Pour chaque point constituant le cluster on cherche ceux qui sont situés à une distance inférieure à ε. Selon le voisinage ces points seront qualifiés de noyaux ou objets limites et seront affectés au cluster A. Ces points sont alors marqués comme "traités" par l'algorithme.
- Les points candidats restants qui ne sont rattachés à aucun cluster et non marqués comme "traités" par l'algorithme sont des "outliers". La méthode DBSCAN est appliquée sur la matrice candidate en $D = 2$ dimensions et $N_c = 100$ points (figure 1.31).

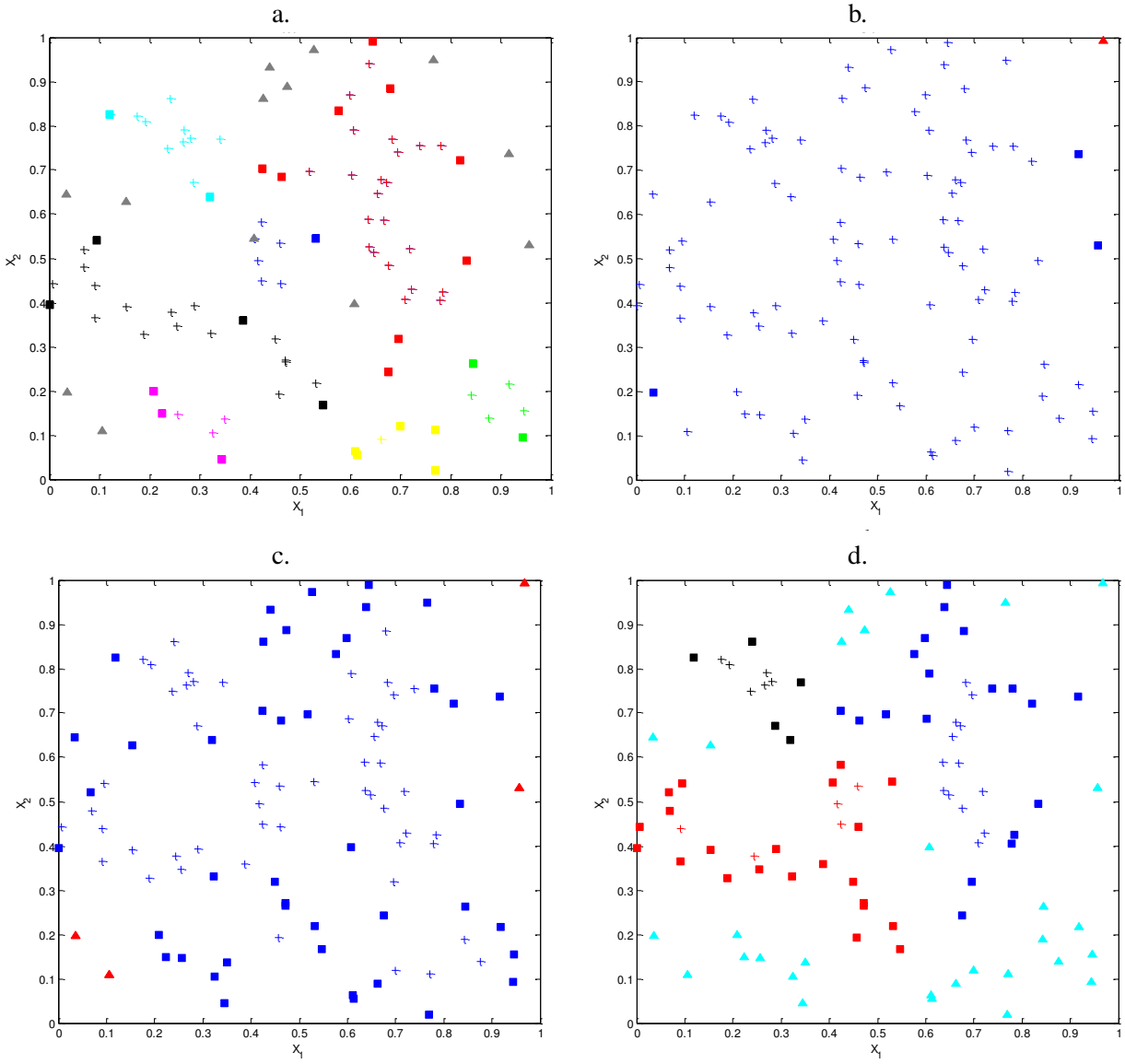


FIGURE 1.31 – Représentation des clusters obtenus par la méthode **DBSCAN** appliquée sur une matrice candidate aléatoire en 2D et $N_c = 100$ points. Chaque couleur correspond à un cluster. $+$: un point noyau, ■ : un objet limite, ▲ : un outlier. **a)** $v = 3$ et $\varepsilon = 0.095$, **b)** $v = 3$ et $\varepsilon = 0.15$, **c)** $v = 5$ et $\varepsilon = 0.122$, **d)** $v = 5$ et $\varepsilon = 0.10$.

Par la méthode DBSCAN, les points sont affectés à une catégorie (noyau, limite, outlier), dépendante du voisinage. Avec la figure 1.31, nous constatons que si nous souhaitons utiliser cet algorithme pour sélectionner des points, nous pourrions proposer de ne retenir que les noyaux dont le nombre dépendra des valeurs des paramètres v et ε. Ainsi, le fait de ne pas pouvoir maitriser le nombre de noyaux rend la méthode difficile pour la sélection de points.

3.1.2.2 Méthode des *k*-means

La méthode des *k*-means [43, 44, 45] est aussi une méthode de clustering c'est-à-dire une méthode destinée à former des clusters de points. Elle divise l'ensemble des données en k clusters, avec k fixé par l'utilisateur. Au début de l'algorithme, les objets sont aléatoirement rattachés aux k clusters. Au cours des itérations les centres de gravité des clusters sont redistribués dans l'ensemble des données afin que les objets similaires appartiennent au même cluster.

Algorithme 3.7 Algorithme *k*-means

Définir le nombre de clusters k à trouver
Affecter aléatoirement les points aux k clusters

JUSQU'A stabilisation du critère E

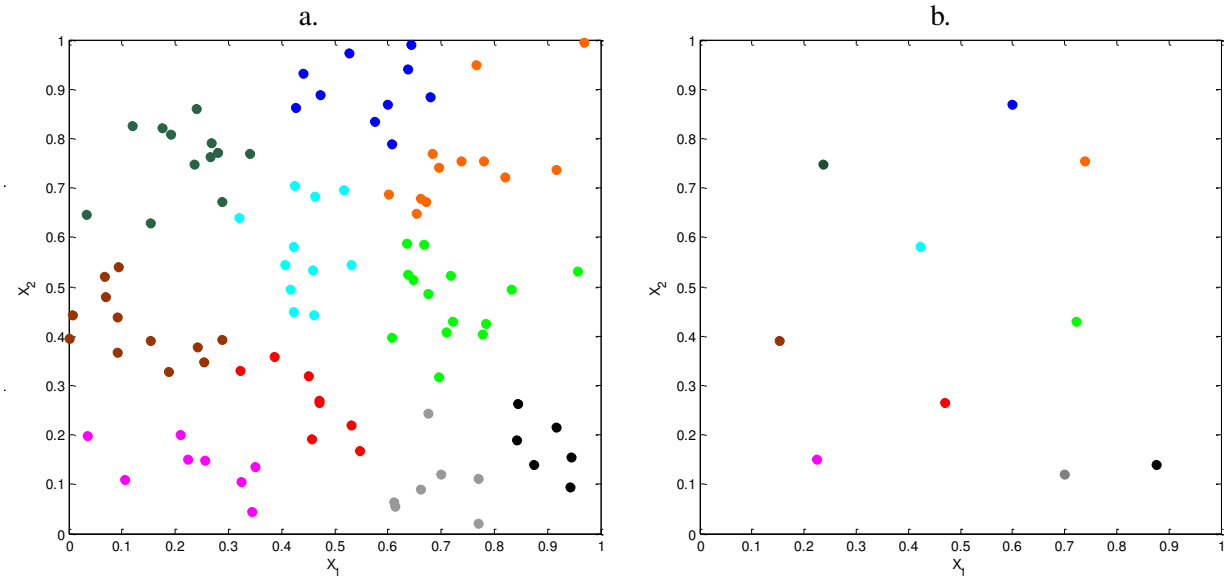
- Calculer le centre de gravité $\bar{x}_j$ de chaque cluster,
- Réaffecter chaque point à son centre de gravité $\bar{x}_j$ le plus proche,
- Calculer le critère E défini par :

$$E = \sum_{j=1}^{k} \sum_{i=1}^{N_c} (x_i - \bar{x}_j)^2$$

avec x_i les coordonnées du point i et $\bar{x}_j$ les coordonnées du centre de gravité de chaque cluster k

FIN

L'algorithme *k*-means permet de regrouper les points dans des clusters en fonction de leur proximité. Si nous souhaitons utiliser cet algorithme pour la sélection de points, nous proposons de sélectionner le point le plus proche du centre de gravité de chacun des clusters construits par l'algorithme *k*-means (figure 1.32). L'utilisation de la méthode des *k*-means pour la sélection de points conduit à des sous-ensembles différents car l'affectation des points aux clusters à la première étape est aléatoire.



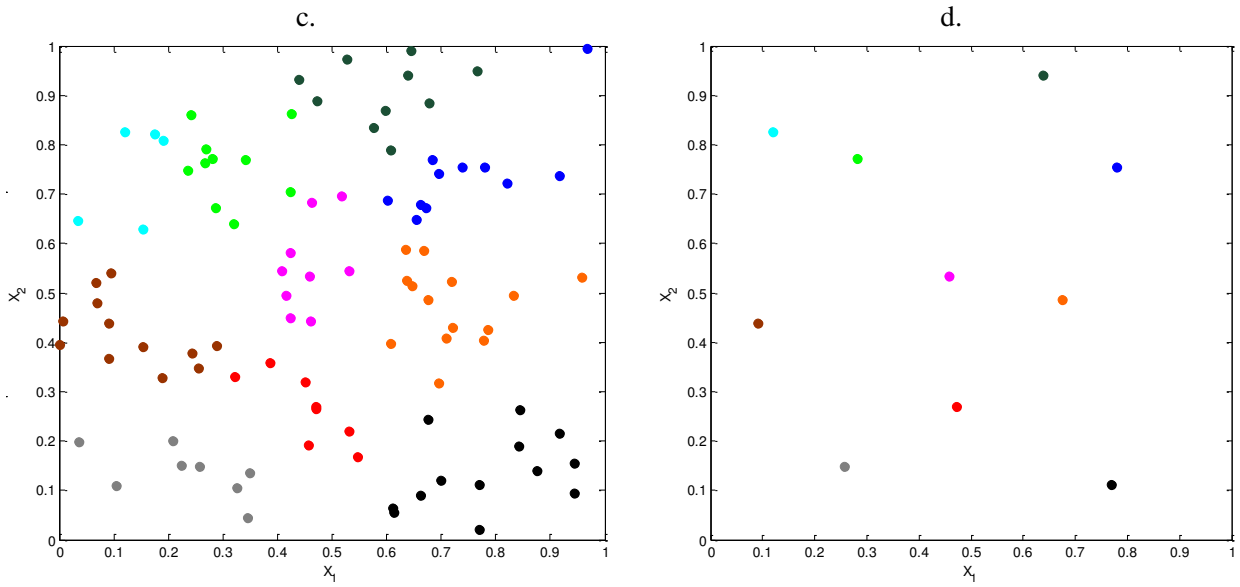


FIGURE 1.32 – Ensembles de points obtenus par l'algorithme ***k*-means**. $k = 10$ clusters sont retenus par la méthode appliquée sur une matrice candidate aléatoire en 2D et $N_c = 100$ points. Pour chaque cluster, différencié par sa couleur, le point sélectionné sera le point le plus proche du centre de gravité (CG) du cluster. L'algorithme débutant par une affectation aléatoire des points aux clusters, l'algorithme est répété. **a)** et **c)** représentent les $k = 10$ clusters, **b)** et **d)** représentent les CG des clusters qui constitueront les points retenus pour former le sous-ensemble.

3.1.3 Comparaison de la qualité des sous-ensembles de points sélectionnés par les différentes méthodes

3.1.3.1 Exemple en deux dimensions

Nous proposons d'utiliser les critères d'uniformité pour comparer la qualité intrinsèque des sous-ensembles et de les représenter pour l'exemple en $D = 2$ dimensions et $N_c = 100$ points présenté ci-dessus dans les diverses illustrations des méthodes de sélection. Nous utiliserons les algorithmes de sélection pour sélectionner $N = 10$ points (figure 1.33) :

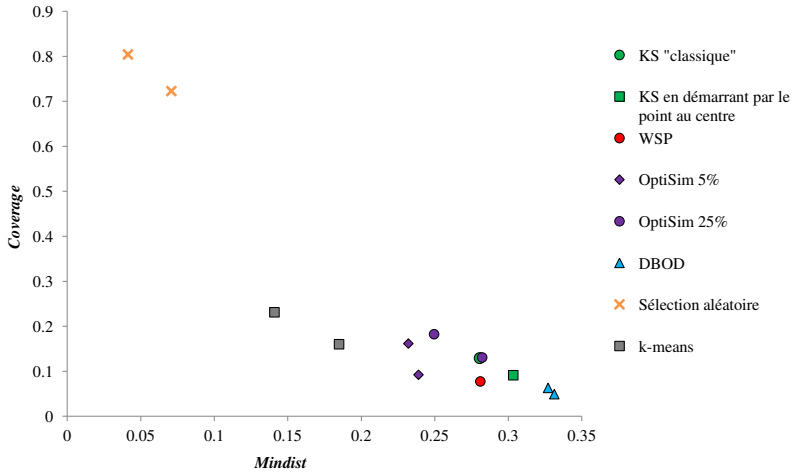
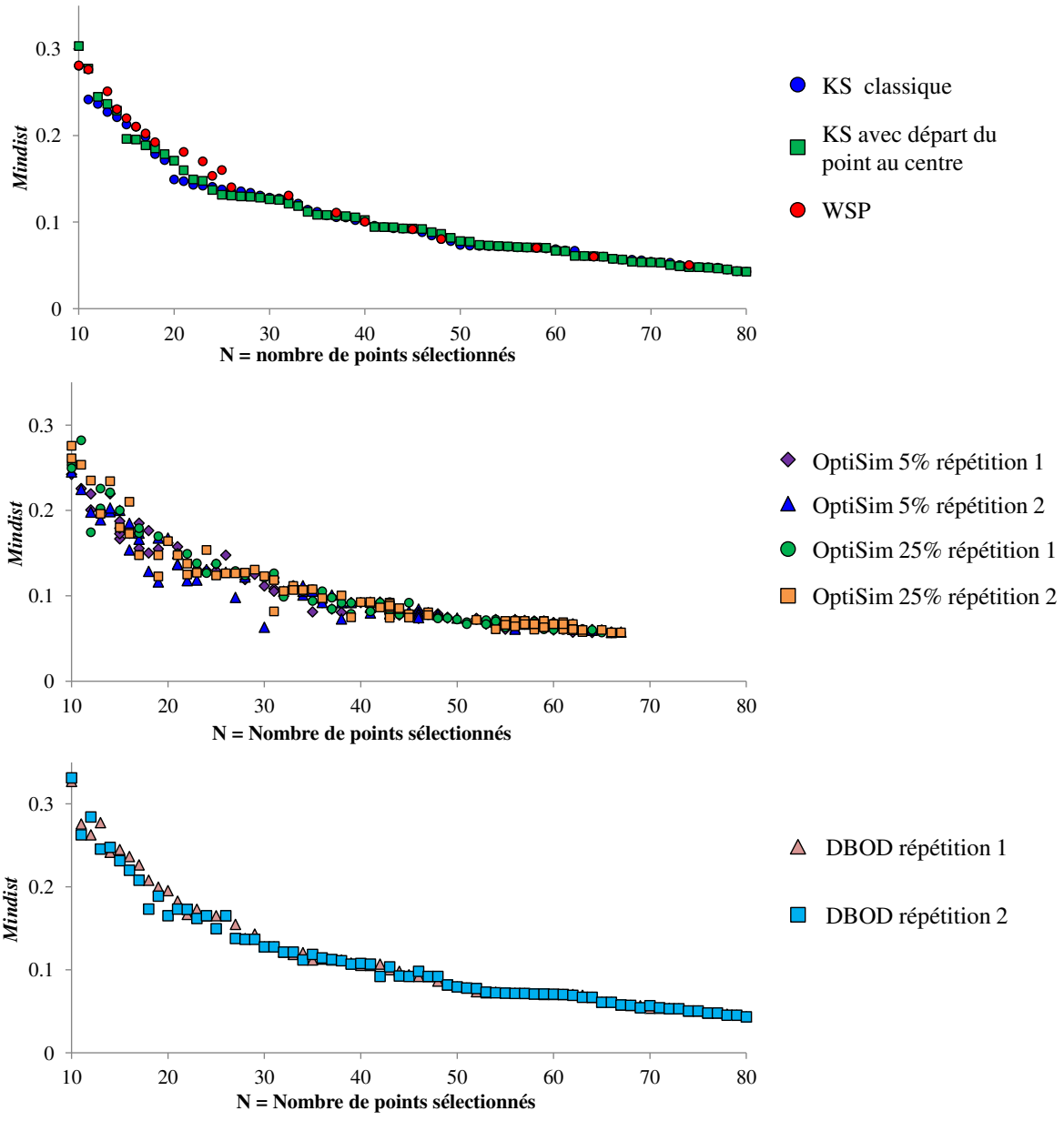


FIGURE 1.33 – Comparaison de la qualité des ensembles à $N = 10$ points sélectionnés par différentes méthodes de sélection à partir d'une matrice candidate aléatoire en $D = 2$ dimensions et $N_c = 100$ points. Les algorithmes qui mettent en jeu un processus stochastique sont répétés deux fois.

La figure 1.33 permet de comparer, pour une valeur de N fixée, la qualité des sous-ensembles obtenus par les différents algorithmes présentés ci-dessus. Nous constatons que les algorithmes reposant sur les distances conduisent aux sous-ensembles de meilleure qualité (valeurs élevées du critère *Mindist* et faibles valeurs *Coverage*). Par ailleurs, les sous-ensembles résultant des algorithmes qui mettent en jeu un processus aléatoire, tels que : OptiSim, *k*-means et la sélection aléatoire, présentent des valeurs de critères différentes lorsque les algorithmes sont répétés pour des mêmes valeurs de paramètres. Seul l'algorithme DBOD permet d'obtenir des ensembles de qualité proche lorsque l'algorithme est répété. Toutefois, si la valeur de N n'est pas fixée nous pouvons alors comparer la qualité des sous-ensembles pour un algorithme donné en fonction de N le nombre de points sélectionnés. Pour cela, nous proposons de suivre et de comparer l'évolution des critères *Mindist* (figure 1.34) et *écart-type* (figure 1.35) en fonction de N le nombre de points sélectionnés par les différents algorithmes, N variant de 10 à 80.



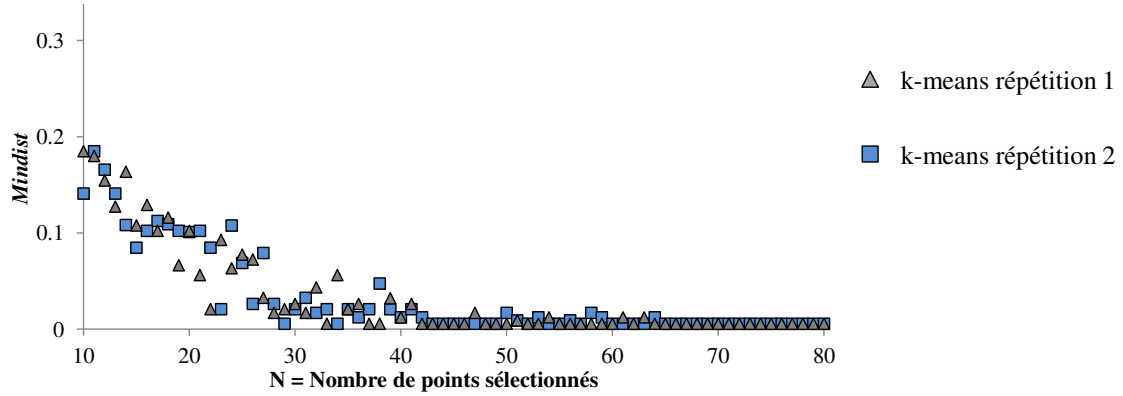
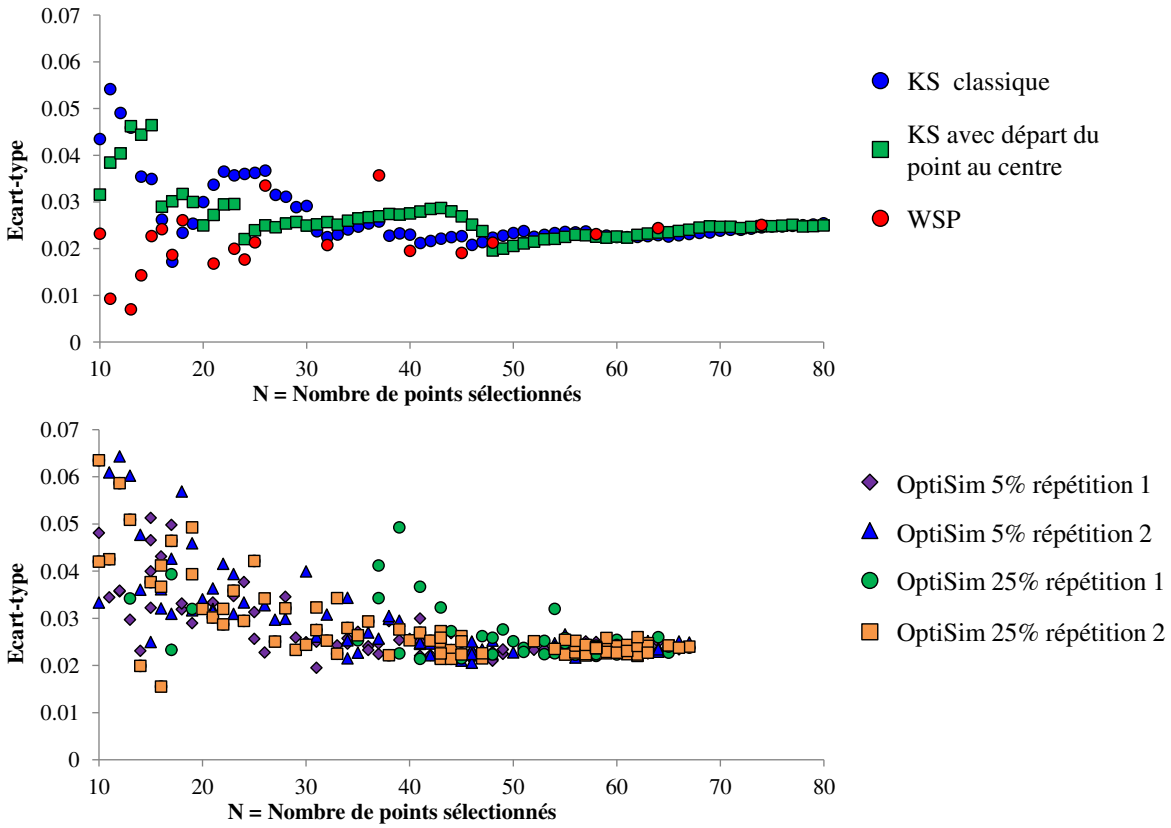


FIGURE 1.34 – Évolution du critère *Mindist* en fonction de N le nombre de points sélectionnés par les différents algorithmes.

La figure 1.34 met en évidence deux comportements pour les valeurs *Mindist* en fonction de la valeur de N. Pour $N < 30$ points, les algorithmes conduisent à des sous-ensembles de qualité différente, avec de fortes variations des valeurs *Mindist* alors que pour $N > 30$ points, les algorithmes conduisent à des sous-ensembles de qualité similaire avec des valeurs *Mindist* très proches. Ainsi, pour $N < 30$ points, les meilleurs sous-ensembles au regard de la distance minimale entre deux points sont obtenus par les algorithmes DBOD, KS et WSP. Les sous-ensembles issus des algorithmes reposant sur un phénomène aléatoire tels que : OptiSim et *k*-means présentent des valeurs *Mindist* plus faibles. Pour $N > 30$ points, les valeurs *Mindist* sont proches signifiant que quel que soit l'algorithme utilisé les sous-ensembles présentent une qualité similaire. Dans une moindre mesure, la méthode des *k*-means adaptée à la sélection de points conduit à des valeurs *Mindist* plus faibles que celles obtenues par les autres méthodes et ce quelle que soit la valeur de N.



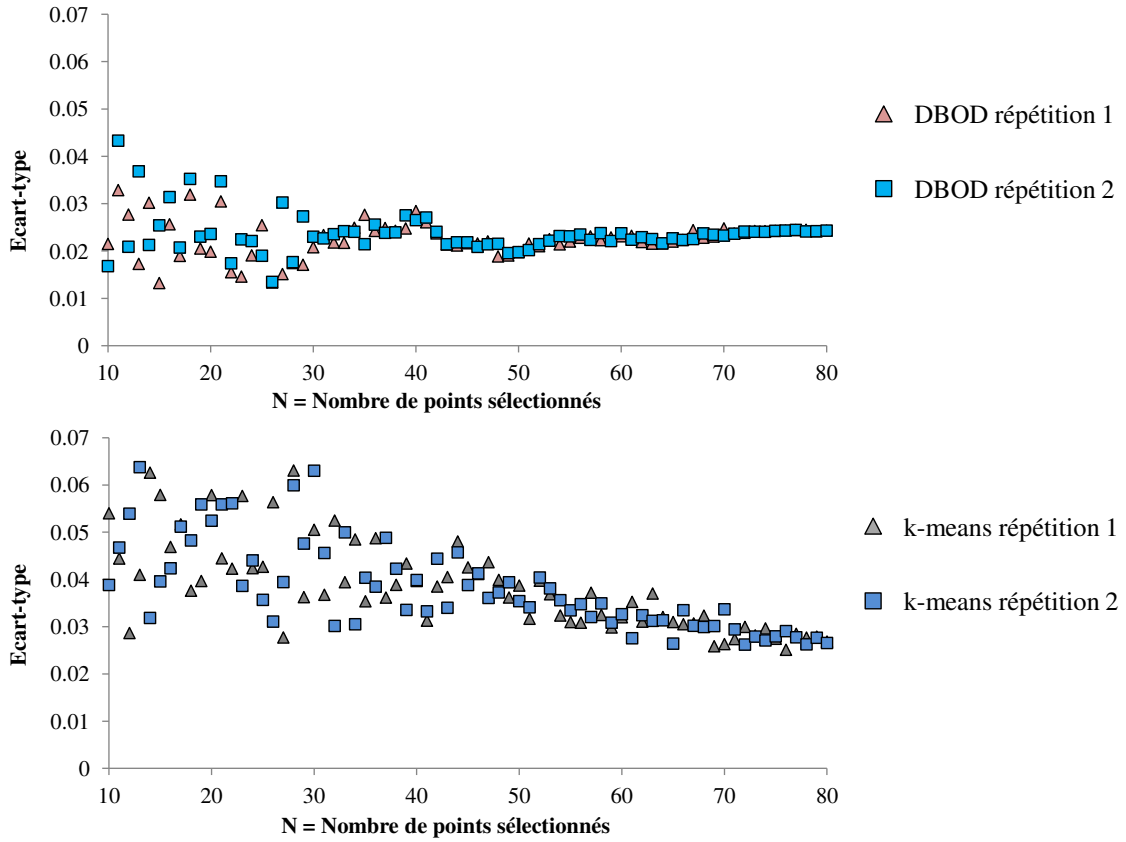


FIGURE 1.35 – Évolution de l'*écart-type* en fonction de N le nombre de points sélectionnés par les différents algorithmes.

A partir de la figure 1.35, nous observons une forte variation des résultats pour les valeurs des écarts-types "inter-algorithmes" mais également "intra-algorithmes" (lors des répétitions d'un même algorithme). Tout comme avec le critère *Mindist*, on observe deux évolutions de l'*écart-type* en fonction des valeurs de N. Pour $N < 30$ points, les méthodes reposant sur le calcul des distances telles que les algorithmes WSP et KS conduisent à de faibles valeurs de l'*écart-type*. Cependant, le point de départ de l'algorithme KS semble avoir une importance car les valeurs fluctuent en fonction de N. Par ailleurs, les sous-ensembles résultant de l'algorithme WSP présentent les valeurs les plus faibles. Quant aux méthodes qui mettent en jeu un processus stochastique, elles présentent une forte variation entre deux répétitions de l'algorithme, justifiant ainsi leur difficulté à être utilisées pour la sélection d'un sous-ensemble comportant peu de points. Pour $N > 30$ points, l'*écart-type* évolue peu et quel que soit l'algorithme utilisé les sous-ensembles obtenus présenteront des qualités similaires.

3.1.3.2 Exemple en 12 dimensions

Nous proposons de réaliser cette même étude pour une matrice aléatoire en $D = 12$ dimensions et $N_c = 600$ points (figure 1.36). Les différents algorithmes de sélection seront alors utilisés pour sélectionner $N = 121$ points.

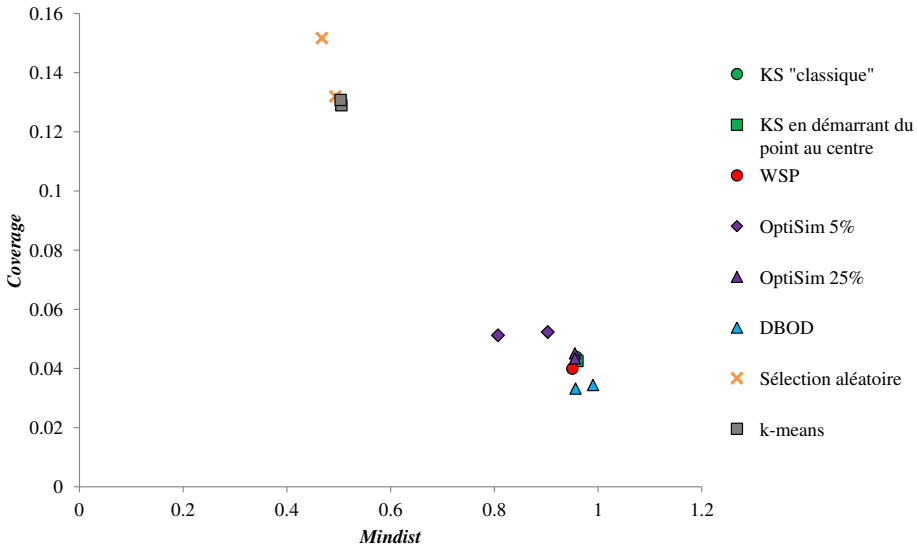
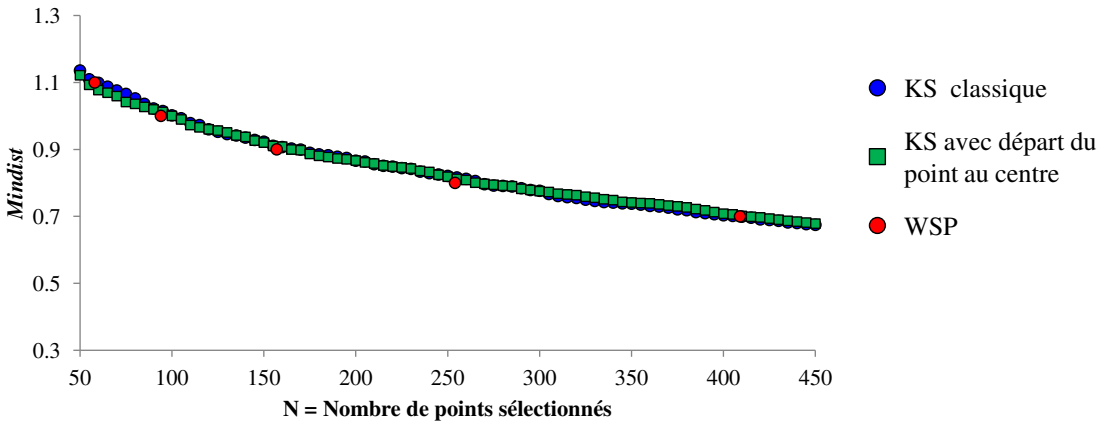


FIGURE 1.36 – Comparaison de la qualité des ensembles contenant $N = 121$ points sélectionnés par différentes méthodes de sélection à partir d'une matrice candidate aléatoire en $D = 12$ dimensions et $N_c = 600$ points. Les algorithmes qui reposent sur un processus stochastique sont répétés deux fois.

A partir de la figure 1.36, nous pouvons généraliser les conclusions faites à partir de l'exemple en deux dimensions. En effet, les méthodes de sélection reposant sur les distances telles que : DBOD, WSP et KS présentent des valeurs *Mindist* et *Coverage* proches signifiant que la répartition des points est de qualité similaire alors que les sous-ensembles résultants de la sélection aléatoire et de la méthode *k*-means sont de moins bonne qualité. Par ailleurs, les algorithmes tels que : OptiSim, DBOD, *k*-means et la sélection aléatoire, qui mettent en jeu un processus stochastique présentent des critères qui varient entre deux répétitions. Dans cette étude seules les méthodes DBOD et *k*-means conduisent à des valeurs de critère très proches après répétition de l'algorithme. Nous proposons de compléter cette étude en suivant l'évolution des critères *Mindist* (figure 1.37) et *écart-type* (figure 1.38) en fonction de N le nombre de points sélectionnés.



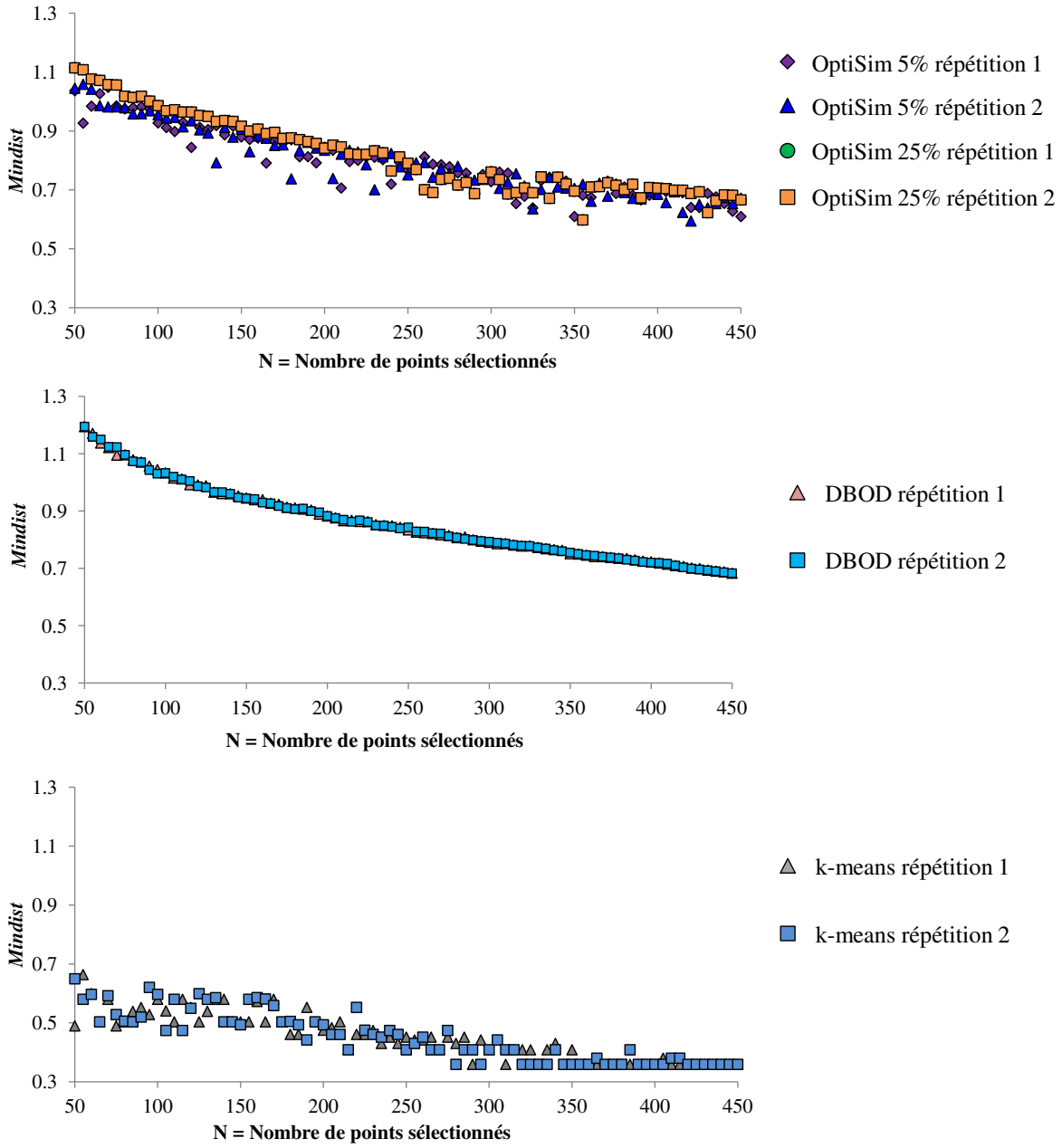


FIGURE 1.37 – Évolution du critère *Mindist* en fonction de N le nombre de points sélectionnés par les différents algorithmes.

La figure 1.37 met en exergue des valeurs *Mindist* plus élevées pour $N < 250$ points pour les algorithmes KS, WSP et DBOD. Les méthodes OptiSim et k-means conduisent à des valeurs plus faibles mais pour $N > 250$ points nous pourrons utiliser indifféremment KS, WSP, DBOD ou OptiSim car le critère *Mindist* prend la même valeur quel que soit l'algorithme utilisé. Seule la méthode des k-means présentent une valeur *Mindist* plus faible et ce pour toutes valeurs de N. Contrairement à l'exemple en deux dimensions (figure 1.34) les répétitions des algorithmes entrainent une fluctuation des résultats beaucoup plus faible.

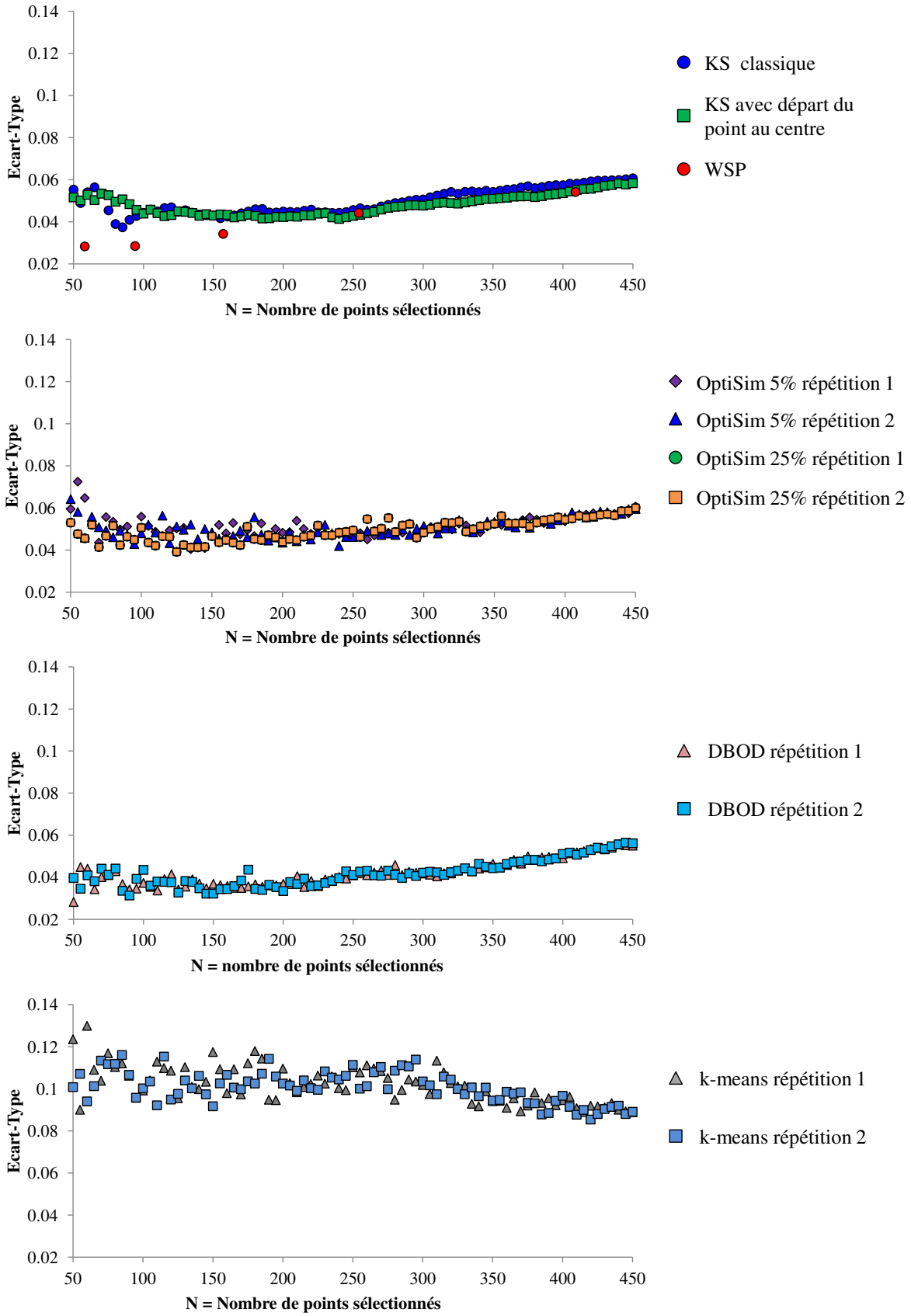


FIGURE 1.38 – Évolution de l'*écart-type* en fonction de N le nombre de points sélectionnés par les différents algorithmes.

A partir de la figure 1.38, nous observons que pour $N < 250$ points les valeurs de l'*écart-type* sont liées à la méthode de sélection. L'algorithme WSP présente les plus faibles valeurs d'*écart-type* pour N petit pour ensuite prendre les mêmes valeurs d'*écart-type* que la méthode KS. Les méthodes DBOD, OptiSim et k-means présentent une fluctuation du critère après répétition de l'algorithme qui s'explique par le phénomène aléatoire mis en jeu par l'algorithme. Pour $N > 250$ points, nous pourrons utiliser n'importe quelle méthode de sélection sauf celle des k-means qui présente des critères bien plus mauvais et ce quelle que soit la valeur de N.

3.1.4 Avantages et inconvénients des méthodes de sélection de points

Quelle que soit la méthode de sélection de points utilisée, l'objectif est d'obtenir un sous-ensemble de N points, avec N inférieur à N_c le nombre de points candidats. Les points sélectionnés doivent apporter une information suffisante et couvrir le domaine expérimental. La démarche proposée par Kennard et Stone (KS) permet de sélectionner les points les plus éloignés les uns des autres ce qui permet de couvrir un large domaine, avec l'objectif d'obtenir une répartition uniforme des points. Le principal inconvénient de la méthode KS est qu'elle ne dispose pas d'un critère d'arrêt. En effet, les différentes étapes de l'algorithme font intervenir un arrangement de N points et permettent ainsi d'obtenir des structures à N points. Toutefois les différentes structures ne permettent pas d'apprécier pour quelle valeur de N points sélectionnés parmi les points candidats, le domaine expérimental sera suffisamment couvert par les points expérimentaux. L'algorithme ne donne pas un critère garantissant des structures optimales que nous appellerons stables c'est-à-dire des structures pour lesquelles les distances séparant un point de tous les autres points sont égales ou très peu différentes. La méthode DUPLEX, qui est une extension de la méthode de KS, permet de construire alternativement les ensembles de calibration et de validation et souffre du même défaut que l'algorithme KS. En effet, les sous-ensembles obtenus auront un large domaine de variation mais aucune information n'est obtenue quant à la bonne répartition des points dans tout le domaine. Par ailleurs, le nombre de points constituant les sous-ensembles de calibration et de validation est nécessairement le même puisqu'il est directement lié à la méthode de construction. Lors de l'étape de modélisation de données spectroscopiques par exemple, les données sont souvent réparties aux $2/3$ pour la calibration et $1/3$ pour la validation, ce qui rend cet algorithme inadapté. La méthode OptiSim permet de simplifier les calculs car l'utilisation des sous-échantillons de points ne nécessite pas de calculer toutes les distances. Toutefois, cette méthode reste compliquée et délicate à mettre en place car l'utilisateur doit fixer la valeur des trois paramètres (ε, k et S) dont les résultats dépendront. Par ailleurs, l'utilisation d'une sélection aléatoire du point appartenant au sous-ensemble final parmi les points du sous-échantillon implique des résultats différents lorsque l'algorithme est répété pour les mêmes valeurs des paramètres. La méthode DBOD a été proposée par Marengo et Todeschini afin de pallier le principal inconvénient de la méthode KS. En effet, l'algorithme procède par substitution d'expériences afin que celles-ci soient les plus éloignées possibles mais seules les substitutions favorables seront réalisées conduisant ainsi à des structures de points plus stables que celles obtenues par KS. Tout comme l'algorithme OptiSim, la méthode DBOD repose sur une sélection arbitraire du premier sous-ensemble conduisant à des solutions différentes lorsque l'algorithme est répété. L'algorithme WSP permet de sélectionner un sous-ensemble de points dans un ensemble de points candidats. Le principal avantage de l'algorithme WSP est la garantie d'obtenir des solutions couvrant uniformément le domaine expérimental. Il présente également l'avantage d'être facile et rapide à mettre en place. Les méthodes

reposant sur les clusters comme k-means peuvent être adaptées pour la sélection de points. A partir du partitionnement des points, nous sommes capables de connaître l'appartenance d'un point à un cluster. Toutefois si nous souhaitons utiliser ces algorithmes, nous pouvons nous demander s'il est judicieux de considérer les centres de gravité des clusters comme points sélectionnés. La méthode du DBSCAN nécessite de fixer deux paramètres (p et ε) dont les résultats dépendront. Dans chaque cluster, un point peut être identifié comme un point limite ou un noyau. Ainsi dans un même cluster il peut y avoir plusieurs noyaux et dans ce cas nous pouvons nous demander si nous devons considérer seulement les noyaux comme points à sélectionner. Si cette solution est retenue alors le nombre de points sélectionnés dans le sous-ensemble ne peut pas être maitrisé. La méthode de Puchwein est une méthode itérative qui peut donc s'avérer lente pour de grandes bases de données. Un autre inconvénient est qu'il est impossible d'obtenir un nombre de points prédéfini. Toutefois parmi les méthodes étudiées ici, la méthode de Puchwein est la seule qui envisage d'utiliser les distances de Mahalanobis, qui dépendent de la matrice de variance-covariance.

Afin de comparer les temps de calcul des différents algorithmes, nous proposons de suivre l'évolution du temps de calcul en fonction du nombre N de points sélectionnés à partir d'une matrice candidate aléatoire en $D = 12$ dimensions et $N_c = 600$ points (figure 1.39). Nous observons que l'algorithme DBOD est le plus lent ce qui peut être expliqué par le calcul de toutes les distances lors de chaque tentative de substitution. L'algorithme de KS est efficace pour de faibles valeurs de N ($N < 250$) puis le temps de calcul augmente lorsque N devient grand. Les résultats des algorithmes WSP, OptiSim et k-means sont obtenus quasiment instantanément ce qui constitue un gros avantage lorsque les données candidates seront de grandes dimensions.

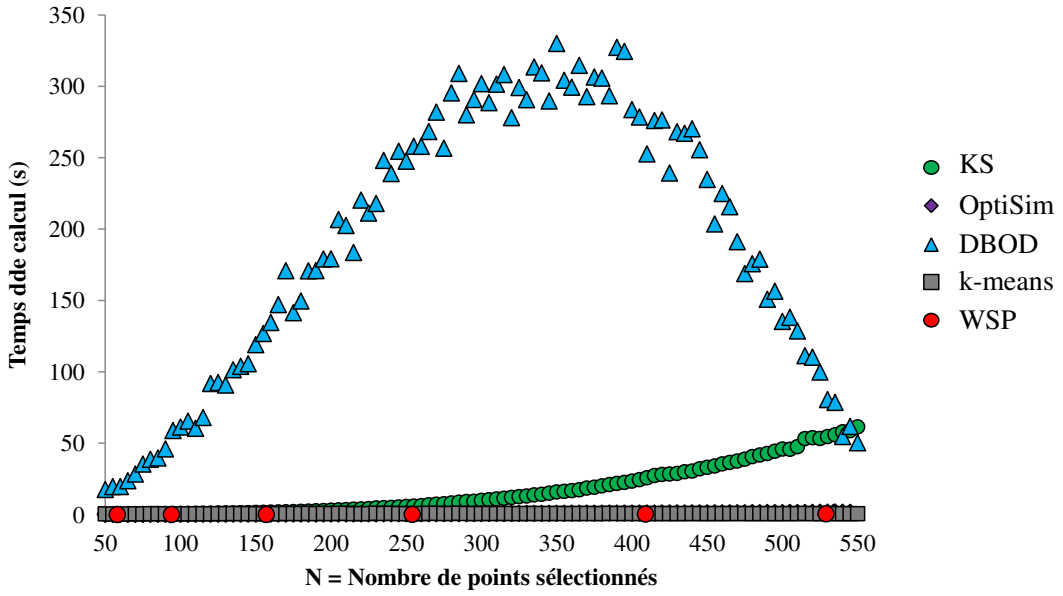


FIGURE 1.39 – Évolution du temps de calcul des différents algorithmes en fonction de N le nombre de points sélectionnés. Les algorithmes de sélection sont appliqués sur une matrice candidate aléatoire en $D = 12$ dimensions et $N_c = 600$ points.

Tableau 1.2 – Tableau récapitulatif des algorithmes de sélection de points.

Méthodes	Paramètres à ajuster	Choix précis du nombre de points N de l'ensemble final	Uniformité de la répartition des N points	Présence de processus aléatoires	Temps de calcul
Kennard et Stone	NON	OUI	NON	NON	assez long
DUPLEX	NON	OUI	NON	NON	assez long
DBOD	NON	OUI	OUI	OUI	très long
OptiSim	OUI (3)	OUI	OUI/NON	OUI	court
WSP	NON	NON	OUI	NON	très court
DBSCAN	OUI (2)	NON	NON	OUI	court
***k*-means**	NON	OUI	NON	OUI	court

Chacune des méthodes pré-citées n'est utilisable que lorsque le domaine est un hypercube. Aussi, après avoir pris en compte les avantages et inconvénients de chacune d'entre elles nous avons choisi de ne retenir que l'algorithme WSP et d'apporter des améliorations afin de pouvoir sélectionner des points dans des domaines quelconques ou de densifier certaines zones présentant un intérêt particulier. Ces avancées sont présentées dans la section suivante.

3.1.5 Les avancées des méthodes de sélection de points

Rappelons que l'objectif de nos travaux est d'utiliser ces algorithmes dans des espaces de grande dimension, pour sélectionner un sous-ensemble de points le plus uniformément répartis. Mais des études plus précises ont montré que les espaces en grande dimension possèdent des propriétés particulières [46, 47] qu'il convient de ne pas ignorer, appelées "fléau de la grande dimension". L'une de ces propriétés consiste en une zone centrale "creuse". En effet, il a été montré que dans une distribution de points en grande dimension, la probabilité qu'ils se situent dans les coins de l'hypercube tend vers 1 quand la dimension augmente. Ceci peut s'expliquer en calculant le ratio des volumes d'un hypercube de dimension D et de l'hypersphère inscrite dans cet hypercube. Le volume de l'hypercube de côté $2r$ et de dimension D est donné par (équation (3.3)) :

$$C_D = (2r)^D \tag{3.3}$$

Le volume de l'hypersphère (équation (3.4)) de rayon r est :

$$S_D = \frac{2^{\lceil (D+1)/2 \rceil} \Pi^{\lceil D/2 \rceil}}{D!!} r^D \tag{3.4}$$

avec $\lceil D/2 \rceil$le plus grand entier inférieur ou égal à $D/2$ et

$$D!! = \begin{cases} \prod_{i=1}^{k}(2i-1), \text{pour } D = 2k-1 \\ \prod_{i=1}^{k}(2i), \text{pour } D = 2k \end{cases} \tag{3.5}$$

On peut voir que ce ratio de volumes (S_D/C_D) tend vers zéro quand la dimension tend vers $+\infty$ (figure 1.40), ce qui signifie que la probabilité qu'un point se situe à l'intérieur de l'hypersphère tend vers zéro.

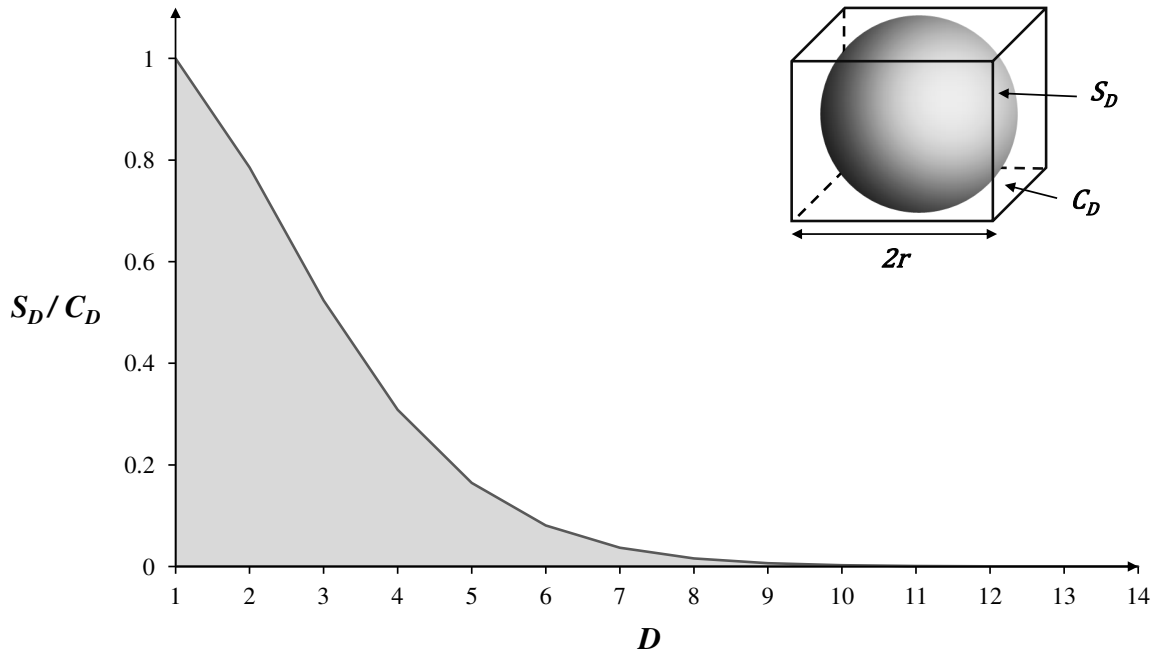


FIGURE 1.40 – Évolution du rapport des volumes de l'hypersphère (S_D) inscrite dans l'hypercube de volume C_D de dimension D.

Ce phénomène peut facilement se vérifier en étudiant la répartition des projections des points d'une distribution quelconque sur les axes factoriels. Nous avons considéré un ensemble de points sélectionné par l'algorithme WSP en 10D et 205 points, qui se veut uniforme par construction. Néanmoins, la représentation graphique des effectifs sur chaque axe factoriel (figure 1.41) montre une surabondance de points dans les intervalles extrêmes et un centre plus "creux". Ces observations nous ont conduit à faire évoluer l'algorithme WSP pour imposer un enrichissement de points au centre du domaine ou plus généralement, dans des zones d'intérêt ou des zones plus denses dont on veut conserver cette particularité.

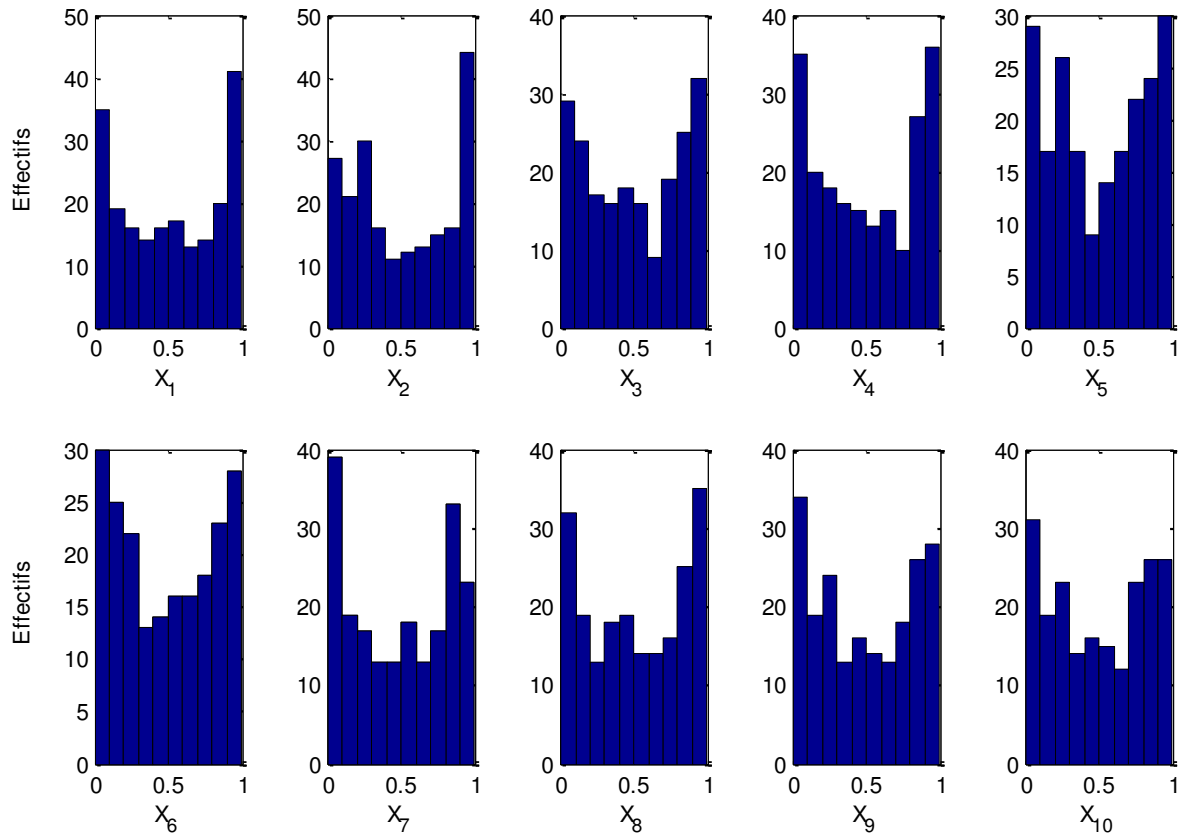


FIGURE 1.41 – Histogrammes représentant les effectifs des points sur les axes factoriels pour un plan construit en utilisant l'algorithme WSP en 10D et 205 points.

Dans l'algorithme WSP, les points étant sélectionnés à partir d'un ensemble de points candidats de telle manière qu'ils soient au moins distants d'une valeur d_{min}, choisie au préalable, des autres points déjà inclus dans le plan, pour densifier une zone de l'espace, on fera varier la valeur d_{min} en fonction de la position du point. Ce nouvel algorithme WSP adapté sera noté **aWSP** (pour adaptive WSP) [48].

3.1.5.1 Densification du centre du domaine

Pour résoudre les problèmes de construction en grande dimension et éviter les centres "creux", nous proposons d'adapter l'algorithme WSP en faisant croître progressivement la valeur d_{min} au fur et à mesure que l'on s'éloigne du centre du domaine. On propose d'utiliser la relation ci-dessous (équation (3.6)) pour calculer la valeur d_{min} à chaque itération en fonction de la position du point :

Soit $X = \{x_1, x_2, ..., x_N\} \subset [0,1]^D$, un ensemble de N points en D dimensions

$$d_{min} = \left[\frac{distance\ entre\ le\ point\ étudié\ et\ le\ centre\ du\ domaine - distance\ minimale}{distance\ maximale - distance\ minimale} \right]^r \tag{3.6}$$

avec $distance minimale = min_{x_i \in X} dist(x_i, x_{centre})$, et $distance maximale = max_{x_i \in X} dist(x_i, x_{centre})$

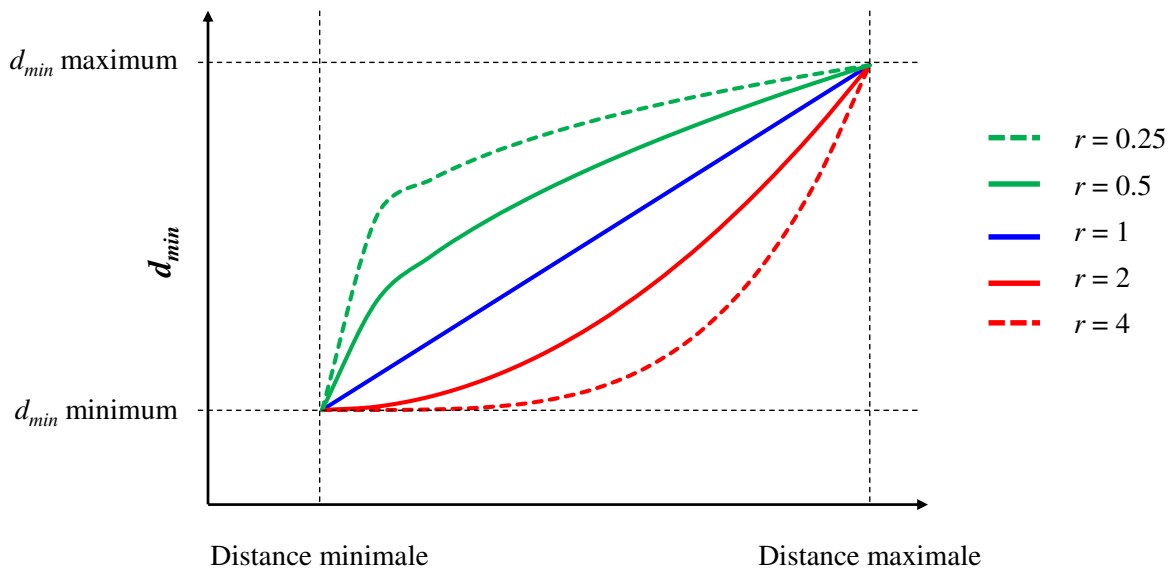


Figure 1.42 – Exemples de fonctions d_{min} pouvant être appliquées à l'algorithme WSP afin de densifier le centre du domaine (r indiquant la valeur de la courbure de la fonction d_{min}).

Lorsque nous souhaitons effectuer une densification du centre du domaine, il est nécessaire de fixer deux valeurs de d_{min}. La première, notée d_{min} minimum, sera appliquée eu centre du domaine, la deuxième, notée d_{min} maximum, sera la valeur d_{min} pour les points situés à la périphérie. La valeur de l'exposant r conditionne la courbure de la variation de la valeur d_{min} (figure 1.42), qui permet d'obtenir une densité de points plus élevée au centre qu'à la périphérie et ce, quelle que soit la valeur de r. Lorsque la courbure r est égale à 1, la valeur d_{min} suivra une variation linéaire. Pour une valeur de r supérieure à 1, la variation de la valeur d_{min} favorisera la sélection de points situés à de faibles distances, c'est-à-dire que la valeur d_{min} augmentera lentement permettant ainsi de conserver davantage de points à proximité du centre du domaine. *A contrario*, une valeur de r inférieure à 1, entrainera une densification très légère du centre du domaine mais sélectionnera moins de points à la périphérie car la courbure de la fonction favorisera très rapidement les grandes valeurs d_{min}. Ainsi, dans l'algorithme WSP, la valeur d_{min} est recalculée à chaque itération en fonction de la distance entre le point considéré et le centre du domaine.

Afin d'illustrer cette méthode de densification du centre du domaine, nous proposons d'appliquer l'algorithme aWSP sur une matrice aléatoire candidate en deux dimensions et 100000 points (figure 1.43). Nous comparerons les matrices obtenues par l'algorithme WSP classique avec une valeur $d_{min} = 0.1$ et par l'algorithme aWSP, pour lequel nous imposons $d_{min}\,minimum = 0.04$ et $d_{min}\,maximum = 0.15$. Nous observerons également l'influence du choix de la courbure sur le nombre de points sélectionnés et leur répartition dans l'espace.

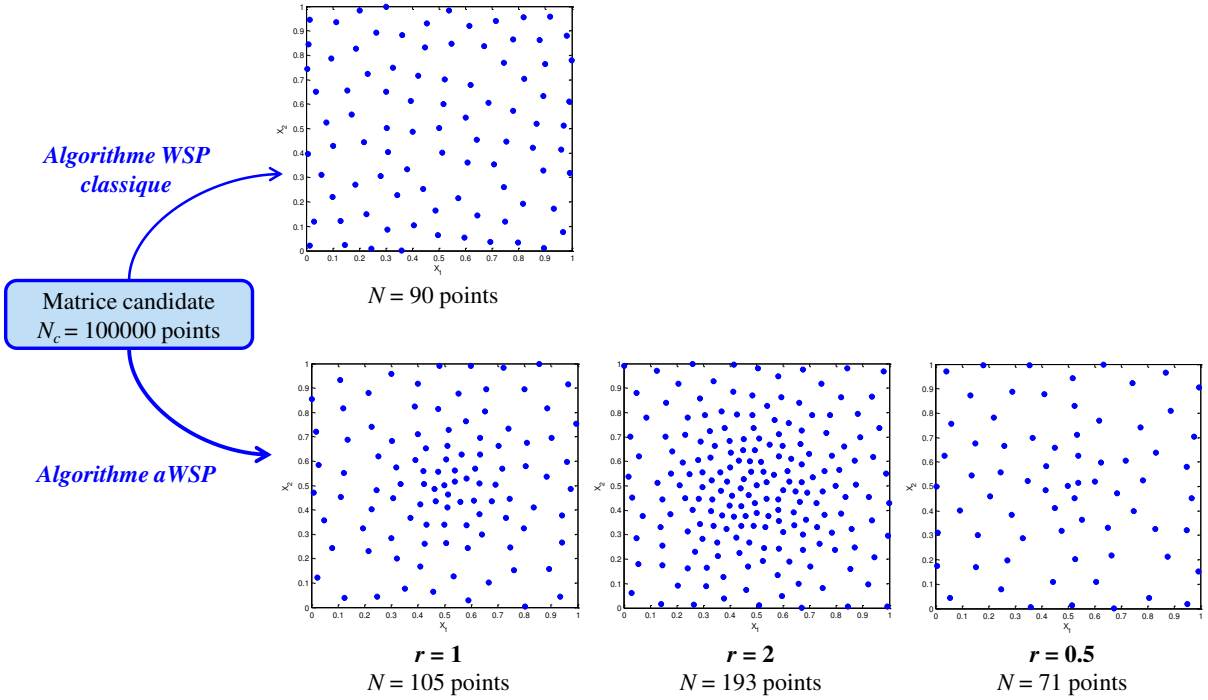


FIGURE 1.43 – Comparaison des algorithmes WSP classique et aWSP pour différentes valeurs de la courbure r.

La figure 1.43 montre la différence entre les deux algorithmes. Alors que l'algorithme classique répartit les points uniformément dans tout le domaine, l'algorithme aWSP va permettre de densifier le centre du domaine et cela en fonction de la valeur du coefficient r qui est choisie par l'utilisateur. Pour $r = 1$, la densification évolue linéairement avec la distance par rapport au centre du domaine, pour $r > 1$, la densification est très importante au centre alors que pour $r < 1$, la densification du centre du domaine est minime et la périphérie s'appauvrit en points. Cet exemple en 2 dimensions est présenté uniquement à titre illustratif car à faible dimension le problème du centre creux ne se pose pas. Cette densification du domaine devient plus intéressante dans des espaces en plus grande dimension. Pour cela, nous proposons deux applications de l'algorithme aWSP en 10 et 20 dimensions.

Exemple en 10 dimensions

Pour montrer les performances de l'algorithme aWSP, nous l'avons appliqué sur une suite de Sobol en 10D et 10000 points, considérée comme matrice candidate en fixant les valeurs de $d_{min}\,minimum = 0.15$, $d_{min}\,maximum = 1.8$ et une courbure $r = 1.9$. La matrice finale compte 205 points, avec un centre plus dense comme nous le montrent les histogrammes des effectifs sur la figure 1.44.

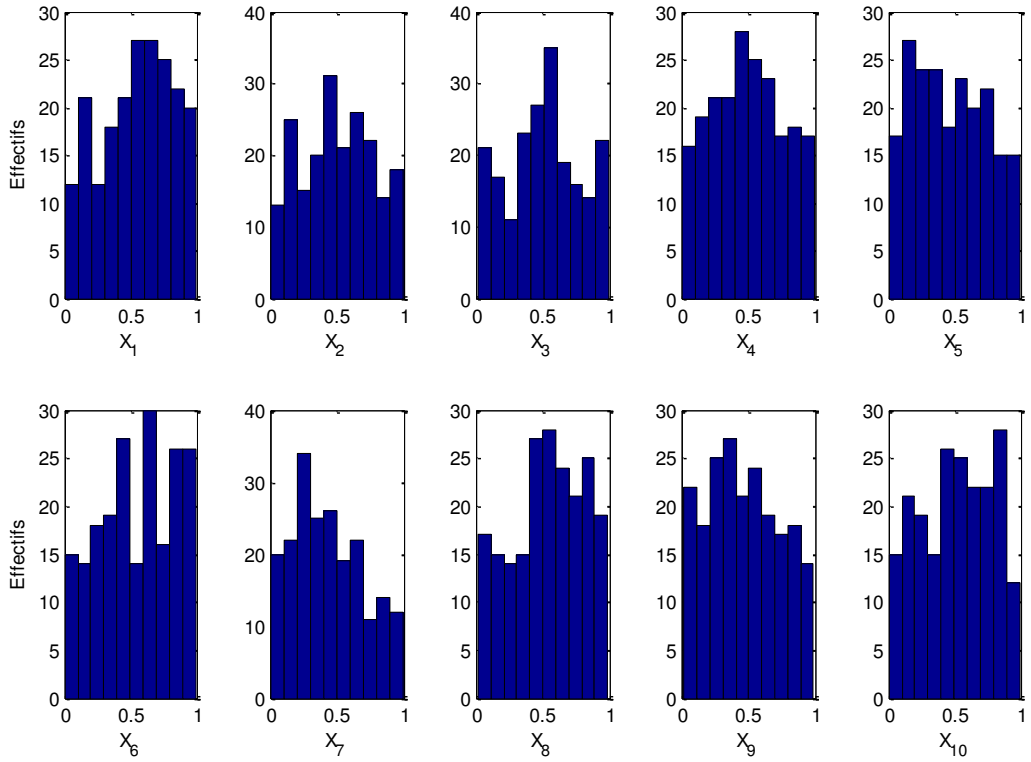


FIGURE 1.44 – Histogrammes représentant les effectifs sur les axes factoriels d'un plan construit en utilisant l'algorithme aWSP avec densification du centre en 10D et 205 points.

La comparaison des histogrammes de la figure 1.41 et de la figure 1.44, montre une modification considérable de la répartition des points dans le domaine des variables. En effet, les effectifs des valeurs centrales sont plus élevés alors que ceux des valeurs extrêmes (à la périphérie du domaine) sont fortement diminués.

Exemple en 20 dimensions

Lorsque la dimension augmente le problème de l'espace vide s'accentue. Pour illustrer la valeur ajoutée de l'algorithme aWSP, nous présentons une application en 20D sur une suite de Sobol avec 20000 points candidats. La représentation graphique de la répartition des points sur les axes factoriels permet de comparer les algorithmes WSP classique et aWSP (figure 1.45). Comme le nombre d'histogrammes est important, seulement quelques uns sont représentés. Ces graphes mettent en évidence les performances de l'algorithme aWSP qui permet d'augmenter le nombre de points au centre du domaine.

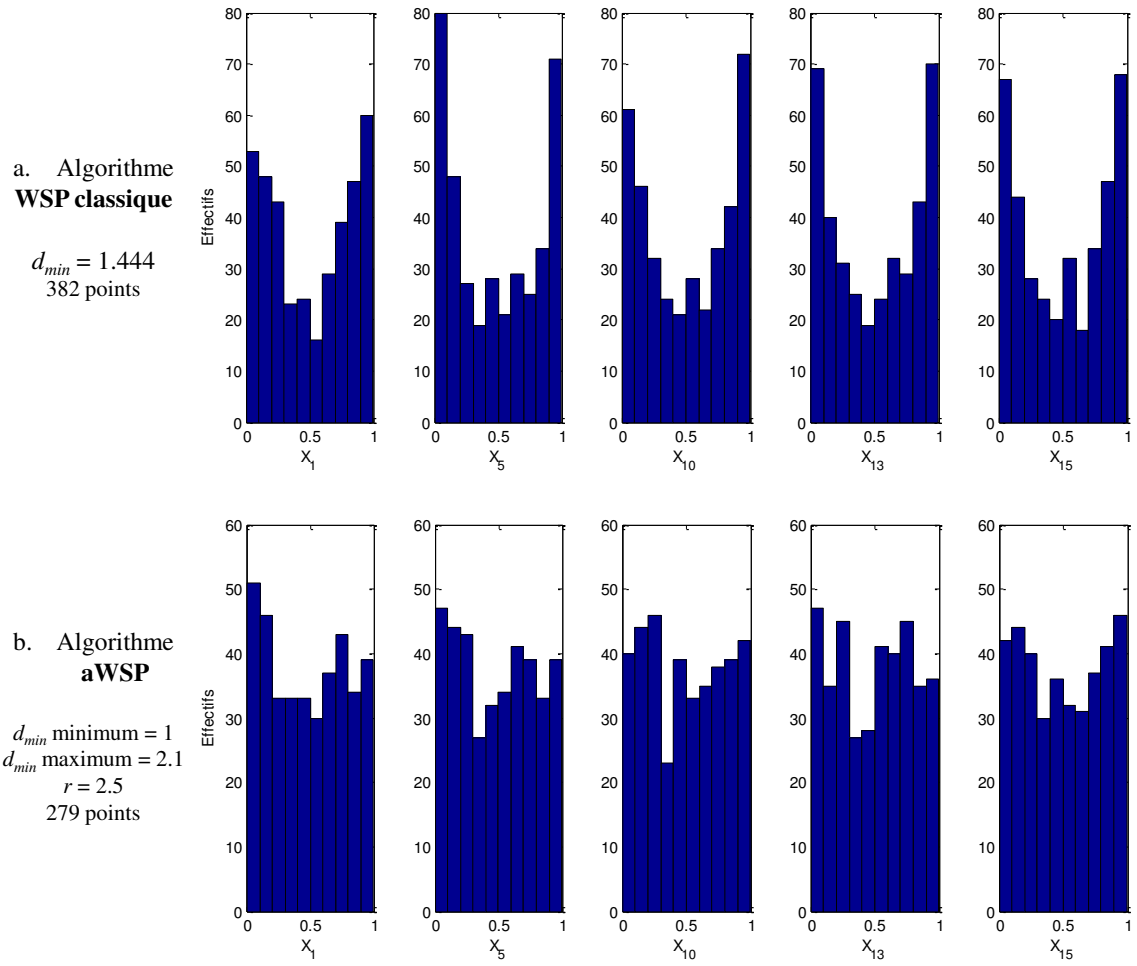


FIGURE 1.45 – Histogrammes représentant le nombre de points sur les axes factoriels pour des plans obtenus **a)** par l'algorithme WSP classique et **b)** par l'algorithme aWSP avec densification du centre du domaine. Ces deux méthodes sont appliquées sur une suite de Sobol en 20D et 20000 points.

Ainsi en utilisant l'algorithme aWSP, nous sommes capables de nous affranchir du fléau de la dimension en densifiant les centres des domaines, et ce quelle que soit la dimension.

3.1.5.2 Densification d'une zone d'intérêt

L'algorithme aWSP peut également être utilisé :

— pour construire des plans uniformes dans des domaines particuliers, lorsque la zone d'intérêt se restreint à une partie du domaine (contraintes de faisabilité, économiques, ...) ,

— pour densifier certaines zones du domaine présentant un intérêt particulier en fonction des connaissances *a priori* du phénomène.

— pour garder la représentativité d'une zone.

Dans ces situations, l'algorithme aWSP utilisera une valeur d_{min} variable en fonction de la position du point dans le domaine expérimental, s'il se trouve dans la zone d'intérêt alors la valeur d_{min} sera plus faible afin de densifier cette zone.

Domaines avec contraintes individuelles

Pour illustrer un domaine avec contraintes, nous considérons une matrice aléatoire en 3D et 100000 points constituant les points candidats. La zone d'intérêt que nous souhaitons densifier est définie par les contraintes suivantes :

$$X_1 \geq 0.6$$
$$X_2 \geq 0.7$$
$$X_3 \leq 0.2$$

La valeur d_{min} appliquée dans la zone d'intérêt est égale à 0.04 et 0.15 dans le reste du domaine, ce qui conduit à la sélection de 524 points (figure 1.46). A l'aide de ces deux représentations, nous observons que la densité de points est plus élevée dans une partie du domaine.

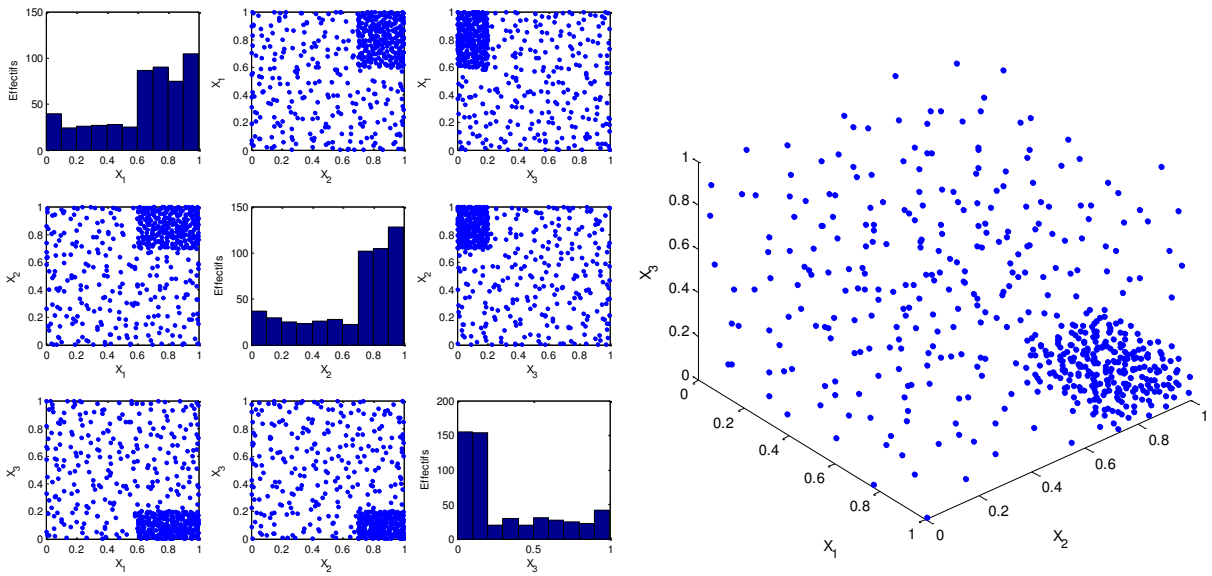


FIGURE 1.46 – Densification d'une zone d'intérêt dans un domaine expérimental à trois variables répondant aux contraintes. A gauche, les différents plans de coupe sont représentés avec sur la diagonale les histogrammes représentant les effectifs sur chaque axe factoriel. A droite, une représentation des 524 points sélectionnés dans une espace en 3D.

Nous proposons de comparer les histogrammes (figure 1.47 a.) des effectifs sur chaque axe factoriel d'une matrice issue d'un WSP classique avec $d_{min} = 0.15$ à ceux de la matrice issue du WSP avec densification de la zone d'intérêt (figure 1.47 b.). Bien que le nombre de points ne soit pas le même dans les deux cas, nous pouvons clairement constater que la zone d'intérêt a été densifiée ce qui se traduit sur les histogrammes par des effectifs plus élevés.

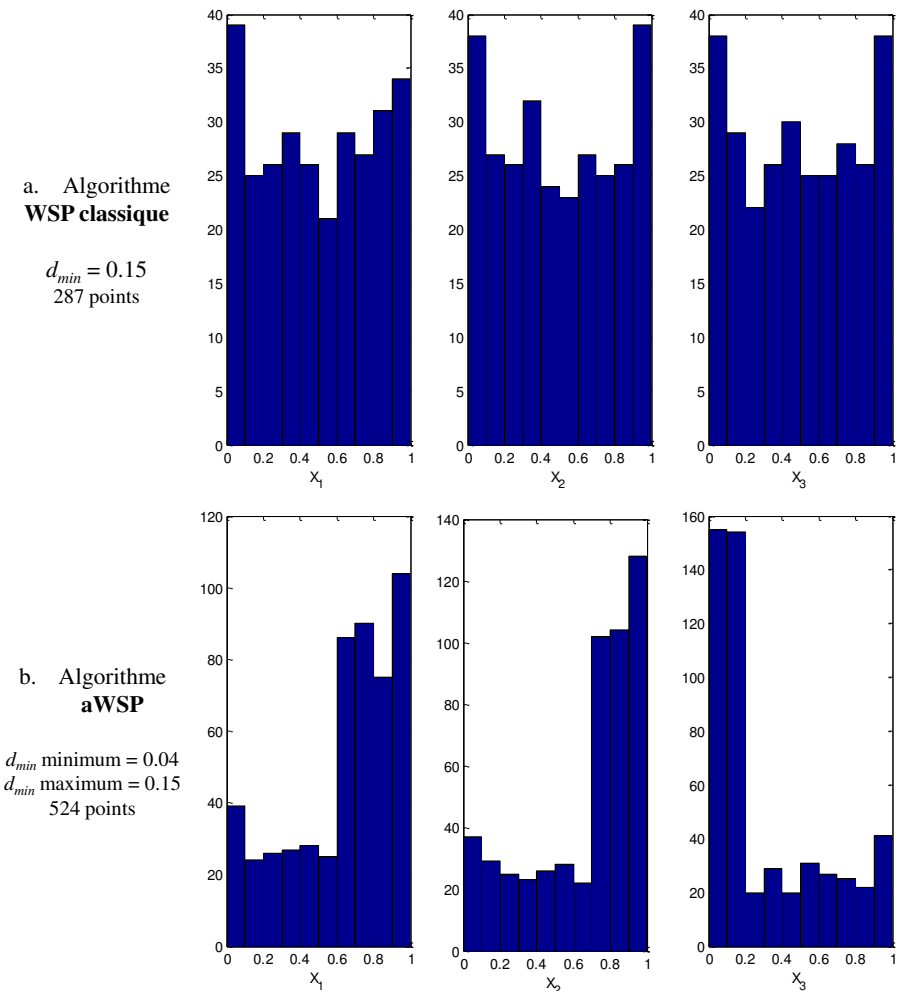


FIGURE 1.47 – Histogrammes représentant les effectifs sur les axes factoriels **a)** des matrices issues d'un WSP et **b)** des matrices issues d'un WSP avec densification, les deux méthodes étant appliquées sur une matrice aléatoire en 3D et 100000 points.

Domaines avec contraintes relationnelles

Les contraintes fixées ici sont des contraintes relationnelles telles que nous pouvons les rencontrer dans des cas d'application réels où les domaines de variation des variables d'entrée sont liés. Les exemples présentés ci-dessous sont en deux dimensions, mais peuvent être généralisés en plus grande dimension.

Lorsque nous souhaitons étudier une zone d'intérêt qui se restreint à une partie du domaine, il est nécessaire de définir les contraintes de la zone à étudier. Pour cela, nous proposons d'appliquer un WSP classique sur une matrice aléatoire en 2D et 100000 points. Nous avons choisi de définir une zone d'intérêt par les contraintes suivantes : $0.5X_1 < X_2 < 0.5X_1 + 0.5$ dans laquelle nous imposons $d_{min} = 0.05$ permettant la sélection de 176 points (figure 1.48 a.).

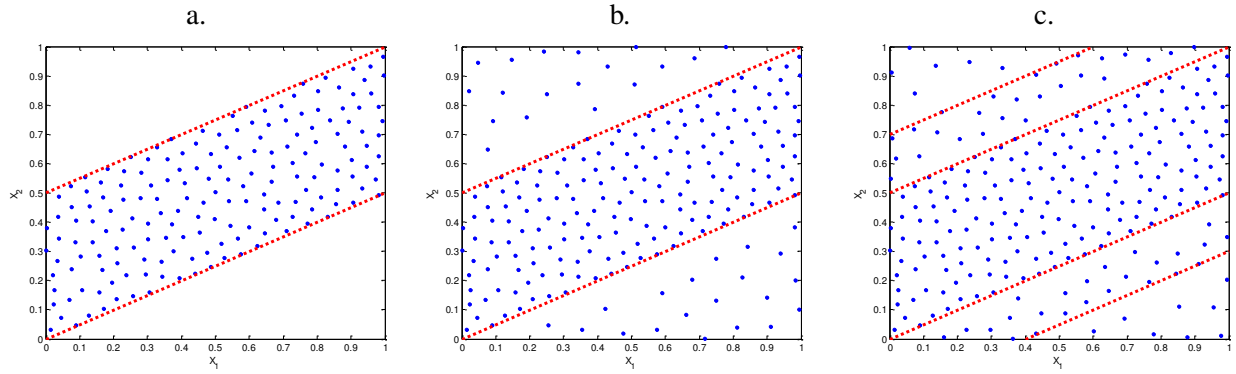


FIGURE 1.48 – **a)** Application de l'algorithme WSP sur une zone avec contraintes : $0.5X_1 < X_2 < 0.5X_1 + 0.5$. **b)** Application de l'algorithme WSP avec une valeur d_{min} plus faible dans la zone d'intérêt permet de sélectionner 214 points, **c)** augmentation de la densité dans la zone d'intérêt avec une évolution progressive de la valeur d_{min} qui sélectionne 240 points.

Dans l'exemple suivant, nous souhaitons étudier le domaine entier tout en densifiant une zone présentant un intérêt particulier contrairement à l'exemple précédent où seule cette zone était envisageable. Pour cela, nous appliquons un algorithme évolutif c'est-à-dire que la valeur d_{min} sera variable dans le domaine. Ainsi, si on considère la même matrice candidate nous définirons une valeur $d_{min}minimum = 0.05$ dans la zone d'intérêt et $d_{min}maximum$ prendra la valeur 0.1 dans le reste du domaine ce qui conduira à la sélection de 214 points (figure 1.48 b.).

Nous proposons également d'appliquer une variation progressive à la valeur d_{min}. La zone à densifier reste inchangée mais nous appliquons une valeur d_{min} intermédiaire à proximité de la zone d'intérêt (figure 1.48 c.). Ainsi, nous définissons trois zones auxquelles nous associons trois valeurs de d_{min}. Dans la zone d'intérêt $d_{min} = 0.05$, pour la partie du domaine entourant cette zone $d_{min} = 0.07$ et $d_{min} = 0.1$ dans le reste du domaine, permettant la sélection de 240 points.

3.2 Les méthodes de sélection de variables

L'objectif d'une analyse de données multidimensionnelles peut aussi être de réduire la dimension par la sélection d'un sous-ensemble de variables, de manière à préserver autant que possible l'information contenue dans l'ensemble initial. La plupart des méthodes proposées dans la littérature utilisent la corrélation linéaire entre variables, ou les valeurs propres obtenues par la décomposition en valeurs singulières ou par les loadings de l'analyse en composantes principales. L'utilisation de techniques basées sur les composantes principales (CP) peut atteindre cet objectif mais les CP sont définies comme des combinaisons linéaires de toutes les variables d'origine, ce qui peut rendre l'interprétation difficile. Aussi, il serait préférable d'utiliser un sous-groupe des variables initiales plutôt que les composantes principales.

Ici, nous proposons une étude bibliographique des principales méthodes de sélection de variables, et nous présenterons plus particulièrement la méthode V-WSP qui est une adaptation et une extension de l'algorithme WSP présenté dans les méthodes de sélection de points.

3.2.1 Les méthodes B2 et B4

Jolliffe a étudié plusieurs méthodes pour la sélection de variables à l'aide des loadings de l'analyse en composantes principales (ACP) [49, 50]. Deux d'entre elles, appelées B2 et B4, nécessitent de sélectionner les variables en fonction de leur assignation aux CP, la différence entre les deux méthodes réside dans le choix de l'ordre des CP les variables aux CP de telle sorte que la variable assignée présente la plus grande valeur absolue du loading.

La méthode B2 est une analyse séquentielle de toutes les CP, qui commence par la dernière c'est à dire par la moins significative. Pour chaque CP, la première variable qui n'a pas encore été choisie et qui présente la plus grande valeur absolue du loading est supprimée. Dans la version non-itérative cette étape est réalisée qu'une seule fois alors que dans la version itérative les composantes principales sont calculées à chaque fois qu'une variable est supprimée du jeu de données. L'idée principale de cette méthode est que les dernières CP apportent l'information la moins pertinente (comme la redondance et le bruit), ainsi les variables qui représentent le mieux ces CP sont celles liées à la redondance et au bruit du jeu de données.

La méthode B4 est également une analyse séquentielle de toutes les CP, qui commence par la première CP. Pour chaque CP, la première variable qui n'a pas encore été choisie et qui présente la plus grande valeur absolue du loading est sélectionnée. Comme les premières CP contiennent la majeure partie de l'information, les variables qui sont les plus représentatives de ces premières CP sont retenues dans le jeu de données.

Le choix du nombre de CP significatives doit être fait au préalable et peut être effectué en utilisant différents critères reposant sur les valeurs propres [51] :

- CAEC (Corrected Average Eigenvalue Criterion) considère comme significatif uniquement les composantes ayant une valeur propre supérieure à la valeur propre moyenne multipliée par 0.7,
- AEC (Average Eigenvalue Criterion) considère comme significatif toutes les composantes avec une valeur propre supérieure à 1.

3.2.2 Le facteur d'inflation *K*

Todeschini et al. ([52]), proposent le facteur d'inflation *K* (*KIF* pour *K* inflation factor) qui est une méthode de réduction de variables reposant sur l'indice de corrélation multivariée *K*. Le principe de cette méthode repose sur l'idée que la structure d'une base de données est le plus souvent conservée lors de la suppression de la variable q telle que les variables restantes présentent une corrélation multivariée minimale. Ainsi, la suppression de la variable q du jeu de données implique que la corrélation multivariée restante est minimale et qu'elle est issue des variables restantes. La valeur KIF_j associée à la j-ème variable est le facteur d'inflation obtenu en considérant la corrélation multivariée totale notée K_p et $K_{p/j}$ l'indice de corrélation multivariée calculé à partir des données en supprimant la j-ème variable. Les auteurs suggèrent de retenir toutes les variables avec une valeur de *KIF* supérieure à la limite proposée égale à 0.5.

3.2.3 La méthode de corrélation par paires

La méthode de corrélation par paires [53] utilise un algorithme simple et reste souvent utilisée dans les études de relations structure-activité. Pour chaque paire de descripteurs corrélés (ou variables), avec un coefficient de corrélation égal ou supérieur à un seuil de corrélation fixé, celle qui présentera la plus grande corrélation avec tous les autres descripteurs, sera supprimée de manière itérative.

3.2.4 La méthode CMC

La méthode CMC (Canonical Measure of Correlation) [54, 55] qui mesure la corrélation entre des ensembles de variables est utilisée pour déterminer le sous-ensemble de variables qui reproduit au mieux les principales caractéristiques de la structure du jeu de données complet. Cette méthode peut être utilisée selon un procédure stepwise qui consiste à comparer chaque variable tour à tour avec l'ensemble des variables ne contenant pas la plus corrélée. Cette étape est répétée de manière itérative en utilisant les variables restantes jusqu'à ce qu'il ne reste que deux variables. A la fin de l'étape d'élimination, les variables peuvent être classées selon leurs valeurs de l'indice CMC et le sous-ensemble de variables avec la plus petite valeur de CMC est inclus dans l'ensemble réduit de variables.

3.2.5 La méthode UFS

La méthode UFS (Unsupervised Forward Selection) est un algorithme de réduction de données qui débute avec les deux variables présentant la plus faible corrélation et sélectionne les variables supplémentaires en fonction de leurs corrélations multiples avec celles qui ont déjà été sélectionnées. L'algorithme s'arrête lorsque la valeur de corrélation de chaque variable restante avec celles qui ont déjà été choisies dépasse un seuil fixé. Ainsi, la méthode UFS permet de sélectionner un sous-ensemble réduit de variables qui sont très proches de l'orthogonalité.

3.2.6 Les méthodes de sélection pas à pas

Ces méthodes de sélection de variables sont utilisées pour choisir le meilleur sous-ensemble de variables explicatives. Parmi ces méthodes, nous pouvons citer :

— la méthode d'élimination progressive ou "backward selection" : l'algorithme débute en considérant toutes les variables dans le modèle. A chaque étape, la variable associée à la plus grande *p*-value est éliminée du modèle, si cette valeur est supérieure au seuil fixé *a priori* (en général 5%). La procédure s'arrête lorsque les variables restant dans le modèle ont toutes une *p*-value inférieure au seuil.

— la méthode d'introduction progressive ou "forward selection" : cette méthode est l'inverse de la méthode "backward". Lors de la première étape, le modèle ne contient aucune variable. A chaque étape, la variable associée à la plus petite *p*-value est ajoutée au modèle, si cette valeur est inférieure au seuil fixé *a priori*. La procédure s'arrête lorsque toutes les variables sont introduites ou lorsque les variables restant dans le modèle ont toutes une *p*-value supérieure au seuil.

— la méthode de régression pas à pas ou "stepwise regression" : à chaque étape de la procédure, on examine à la fois si une nouvelle variable doit être ajoutée selon un seuil d'entrée fixé, et si une des variables déjà incluses doit être éliminée selon un seuil de sortie fixé. Cette méthode permet de retirer du modèle d'éventuelles variables qui seraient devenues moins indispensables du fait de la présence de celles nouvellement introduites. La procédure s'arrête lorsque aucune variable ne peut être rajoutée ou retirée du modèle en fonction des critères choisis.

3.2.7 Algorithme V-WSP

L'algorithme WSP a été développé pour sélectionner un sous-ensemble de points dans un ensemble de points candidats afin que ces derniers soient uniformément répartis dans l'espace des variables avec une information de bonne qualité.

Lors des études considérant un grand nombre de variables (comme les études QSAR par exemple qui mettent en jeu un nombre important de descripteurs), il peut être nécessaire de supprimer les variables corrélées. Ainsi, l'algorithme WSP a été adapté pour sélectionner non plus des points mais un ensemble de variables représentatives de l'ensemble initial. Les variables sont choisies en fonction de la valeur minimale fixée du coefficient de corrélation (*thr*) pour toutes les variables définissant l'espace multidimensionnel. Cet algorithme appelé V-WSP permet de réduire le nombre de variables en fonction de la corrélation linéaire existant entre celles-ci. L'objectif de la réduction de variables par l'algorithme V-WSP n'est pas de conserver la structure exacte des données initiales mais d'éliminer l'information redondante. L'algorithme V-WSP peut se résumer ainsi (Algorithme 3.8) :

Algorithme 3.8 Algorithme V-WSP

Considérer un ensemble de N_c points dans l'espace en D dimensions
Choisir une variable de référence i et une limite de corrélation (thr)
JUSQU'À ce que toutes les variables soient sélectionnées ou éliminées

- Calculer les coefficients de corrélation linéaire de Pearson (c) entre la variable i et toutes les autres variables
- Éliminer les variables j telles que : $|c_{ij}| \geq thr$
- La variable i est sélectionnée et remplacée par la variable (parmi les variables restantes) présentant la plus grande valeur absolue du coefficient de corrélation avec la variable i

FIN

L'algorithme V-WSP nécessite de fixer au préalable une limite de corrélation thr et une variable de référence bien qu'il ait été montré que l'algorithme converge vers la même solution quelle que soit la variable de référence fixée.

Le coefficient de corrélation c (équation (3.7)) caractérise la covariance entre deux variables i et j (avec $i \neq j$) rapportée au produit de leurs écarts-type.

$$c = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{3.7}$$

La comparaison de la similarité de l'information contenue dans le jeu de données entier (contenant toutes les variables) par rapport au sous-ensemble de variables réduit est quantifiée par le critère *procrustes*. L'analyse procrustéenne [56, 57, 58, 59] est une méthode statistique utilisée pour comparer la forme entre deux structures : la première est utilisée comme référence et la seconde est déformée par transformations linéaires telles que la translation, la rotation ou une mise à l'échelle afin de la faire coïncider au mieux à la première structure. Dans le cas de la réduction de variables par V-WSP, le critère *procrustes* est calculé à partir des scores ACP du jeu initial contenant toutes les variables et du jeu réduit. Le critère *procrustes* se définit comme la somme des erreurs au carré, que l'on cherche à minimiser afin de conserver au mieux l'information initiale lors de la réduction de variables (équation (3.8)).

$$procrustes = \begin{cases} 0, \text{ les structures sont similaires} \\ 1, \text{ les structures sont différentes} \end{cases} \tag{3.8}$$

Dans un premier temps, nous proposons d'appliquer l'algorithme V-WSP sur le jeu de données classique Aphid proposé par Jeffers [60] qui mesure 19 variables sur 40 pucerons ailés constituant les échantillons.

Dans cet exemple, nous fixons une valeur $thr = 0.85$ et nous changeons la variable de référence, ce qui permet d'obtenir autant de solutions que de variables, ici nous obtenons 19 solutions. Le changement de la variable de référence modifie à la fois le nombre et les variables sélectionnées par l'algorithme V-WSP dans les solutions. Ainsi, pour une solution obtenue pour une variable de référence donnée, nous calculons le critère *procrustes* (figure 1.49) sur les scores ACP des 4 premières composantes principales.

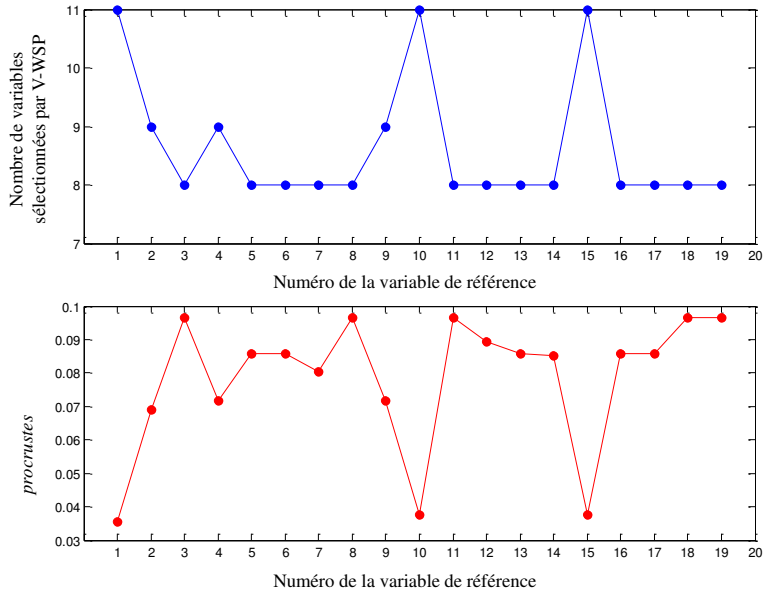


FIGURE 1.49 – Évolution du nombre de variables sélectionnées par V-WSP et du critère *procrustes* pour les 19 solutions obtenues en fonction de la variables de référence choisie.

La figure 1.49 met en évidence des solutions comptant entre 8 et 11 variables sélectionnées par V-WSP. Par ailleurs, la valeur du critère *procrustes* semble être dépendante du nombre de variables dans la solution étudiée, même si le calcul de ce critère est effectué sur les scores des 4 premières composantes principales de l'ACP. A partir de cette observation, il serait facile de conclure que les meilleures solutions au regard du critère *procrustes* sont celles à 11 variables. Or, quelle que soit la solution envisagée, la valeur du *procrustes* reste faible, inférieure à 0.1, mettant ainsi en exergue une forte similarité entre la structure initiale à 19 variables et les solutions V-WSP, démontrant que les solutions sont équivalentes quelle que soit la variable de référence.

A ce stade de l'étude, il est intéressant de connaitre les variables les plus explicatives. Pour ce faire, nous représentons les effectifs des variables pour les 19 solutions (figure 1.50). Nous observons que 5 variables (les variables 5, 11, 17, 18 et 19) apparaissent dans toutes les solutions ce qui permet de conclure sur le fort pouvoir explicatif de ces 5 variables.

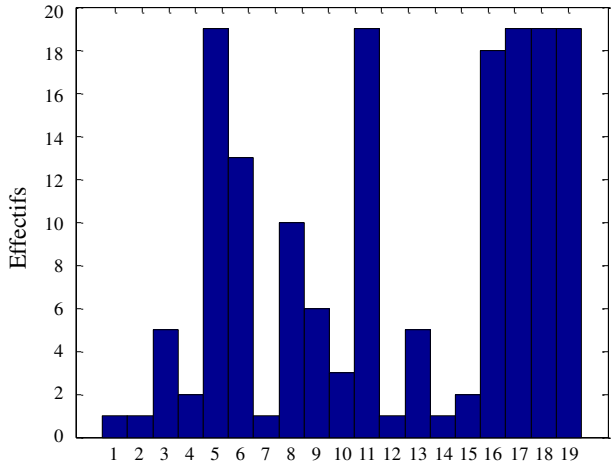


FIGURE 1.50 – Histogramme représentant les effectifs des variables obtenues dans les 19 solutions en fonction de la variable de référence fixée.

La méthode V-WSP permet de sélectionner les variables les plus explicatives tout en éliminant les variables corrélées ou redondantes. Dans la deuxième partie de ce manuscrit, nous discuterons de l'importance du choix de la limite de corrélation *thr* sur les solutions et sur la valeur du critère *procrustes*, à partir de cas d'application.

3.3 Conclusion sur les méthodes de sélection

Nous avons montré qu'une des solutions pour faciliter l'interprétation des données en grande dimension est l'utilisation d'outils permettant de réduire la dimensionnalité en sélectionnant soit des points (lignes) soit des variables (colonnes).

Après avoir comparé les méthodes de sélection de points citées dans la littérature, nous avons choisi de retenir principalement l'algorithme **WSP** qui nous permet de sélectionner des points en garantissant la représentativité de l'ensemble initial. Puis, nous avons proposé une adaptation de cet algorithme, appelée **aWSP** pour résoudre le problème des centres "creux" ou pour prendre en compte des phénomènes particuliers en privilégiant des zones d'intérêt. Les différents exemples présentés ont montré que ce nouvel algorithme est très utile lorsque la dimension augmente.

Parmi les méthodes de sélection de variables, nous retiendrons l'algorithme **V-WSP** pour extraire un sous-ensemble de variables représentatives de l'ensemble initial en supprimant l'information redondante.

Chapitre 4

Le reconditionnement

Nous avons déjà évoqué le fait que toutes les structures de points ne présentent pas la même qualité et que cette différence s'accentue lorsque la dimension augmente. Le mauvais conditionnement d'une base de données ou d'un plan d'expériences, au sens d'une répartition non uniforme des points dans l'espace des variables, sera synonyme entre autres de présence d'amas et/ou de lacunes.

Dans ce chapitre, nous présenterons des méthodes de "réparation" pour des ensembles de points répartis non uniformément dans le domaine des variables. La première étape de la réparation consistera à utiliser l'algorithme de sélection WSP pour supprimer les amas et la seconde étape permettra de combler les zones lacunaires. Nous proposons de suivre les deux étapes de cette réparation à partir d'un plan aléatoire en deux dimensions avec 100 points qui présente des amas et des lacunes.

4.1 Élimination des amas

La première étape de la réparation de plans peut consister à éliminer les amas, constitués de points très proches dans l'espace des variables. En effet, une accumulation de points en certaines zones de l'espace représente une surabondance d'information dans ces régions, qui n'est pas toujours souhaitée initialement. L'utilisation de l'algorithme WSP pour cette étape nous a semblé opportune puisque cet algorithme de sélection est basé sur des calculs de distance et garantit une distance minimale, d_{min}, entre les points. La difficulté réside au niveau du choix de la valeur d_{min} qui va conditionner la distance à partir de laquelle nous définissons un amas.

Pour illustrer cette étape de suppression des amas, nous considérons un ensemble de 100 points dans un espace en deux dimensions, qui constituera notre plan de départ représenté par la figure 1.51.

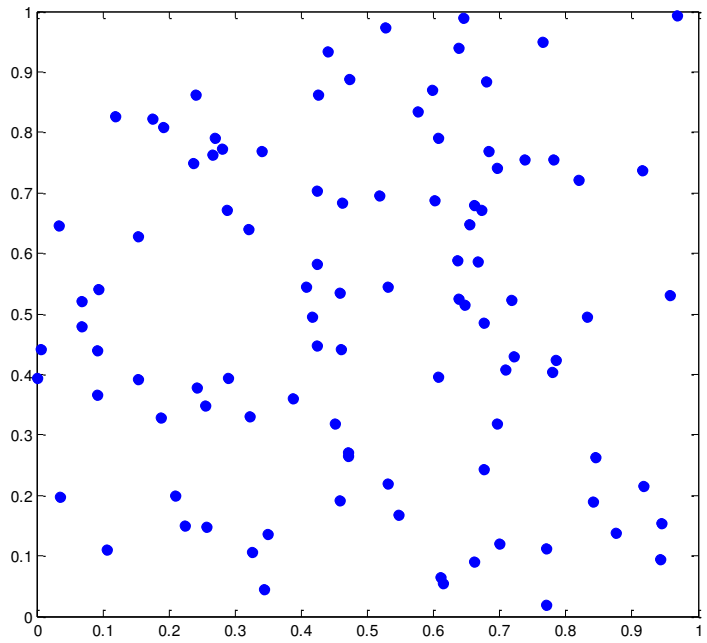


FIGURE 1.51 – Plan en deux dimensions et 100 points.

Sur la figure 1.51, nous observons que ce plan est mal conditionné car il présente des amas et des lacunes et donc de mauvais critères intrinsèques : une valeur *Mindist* faible ($= 0.006$) caractéristique de la présence d'amas et une valeur *Coverage* élevée ($= 0.580$) caractéristique d'une hétérogénéité des distances. Dans un premier temps, nous souhaitons supprimer les amas et pour cela nous proposons d'utiliser l'algorithme WSP qui nécessite de fixer une valeur d_{min}. Tout d'abord, nous allons choisir trois valeurs d_{min}, ce qui nous permet de visualiser l'impact du choix de la valeur d_{min} sur le nombre de points sélectionnés par l'algorithme WSP. La figure 1.52 représente les sous-ensembles de points sans amas résultant de l'algorithme WSP pour des valeurs d_{min} égales à 0.05, 0.10, 0.15 et 0.20 et qui comptent respectivement 74, 40, 24 et 17 points.

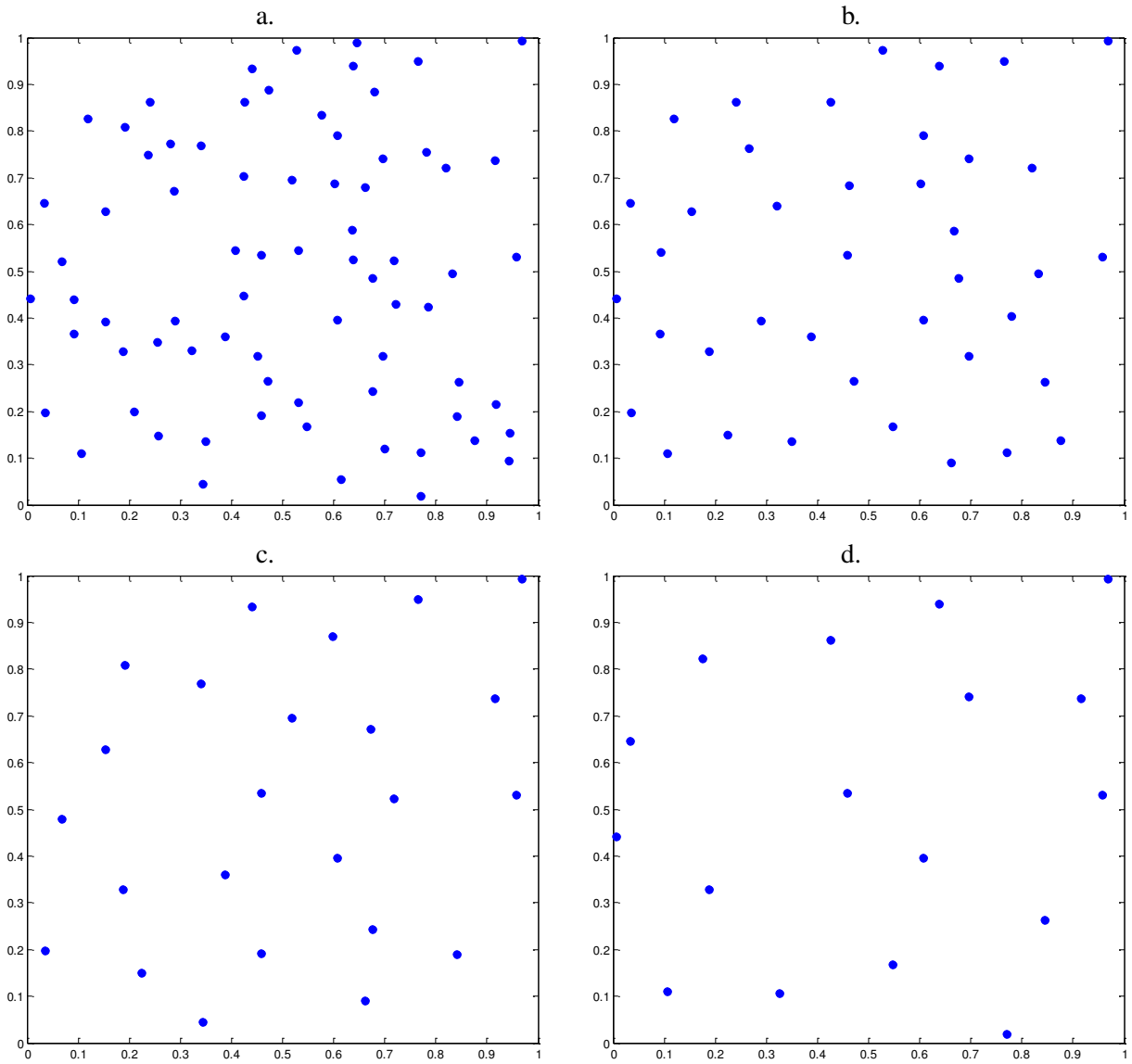


FIGURE 1.52 – Suppression des amas à partir d'une matrice en deux dimensions avec 100 points pour différentes valeurs d_{min}. **a)** $d_{min} = 0.05$ permet de sélectionner **74** points, **b)** $d_{min} = 0.1$ permet de sélectionner **40** points, **c)** $d_{min} = 0.15$ permet de sélectionner **24** points, **d)** $d_{min} = 0.2$ permet de sélectionner **17** points.

La figure 1.52 montre que le choix de la valeur d_{min} conditionne le nombre de points restants après l'étape de suppression des amas. Pour déterminer cette valeur d_{min} nous choisissons de ne pas procéder par itération mais de nous référer à une distribution uniforme de points de même dimension avec le même nombre de points, construit par l'algorithme WSP. Les critères intrinsèques de cette distribution de référence nous permettront de fixer la valeur d_{min} (égale au critère *Mindist*) et ainsi de connaître la distance minimale entre deux points. Nous posons alors que deux points distants d'une valeur plus faible que la valeur d_{min} seront considérés comme deux points proches et formeront un amas.

L'algorithme de suppression des amas peut être écrit ainsi (Algorithme 4.1) :

Algorithme 4.1 Algorithme de suppression des amas

Générer un plan de référence dans la même dimension et avec le même nombre de points que le plan à réparer

Calculer la valeur *Mindist* du plan de référence

Choisir le point initial de l'algorithme WSP

Appliquer l'algorithme WSP avec comme valeur d_{min} la valeur du *Mindist* de la distribution de points de référence

Pour supprimer les amas du plan de départ en 2D et 100 points, nous devons construire en parallèle une distribution de 100 points répartis uniformément dans un espace en 2D. Ce plan constituera le plan de référence pour lequel la distance minimale entre deux points est égale à 0.094. Cette valeur définira la valeur d_{min} de l'algorithme WSP que nous appliquerons sur l'ensemble de points "à réparer" en fixant que deux points dont la distance est inférieure à 0.094 sont considérés comme trop proches et constituent un amas. Ces points seront alors éliminés. Cette étape conduit à l'élimination de 57 points et donc à n'en conserver que 43. Les critères intrinsèques de ces deux plans sont reportés dans le tableau 1.3 et la répartition des points les constituant est représentée sur la figure 1.53.

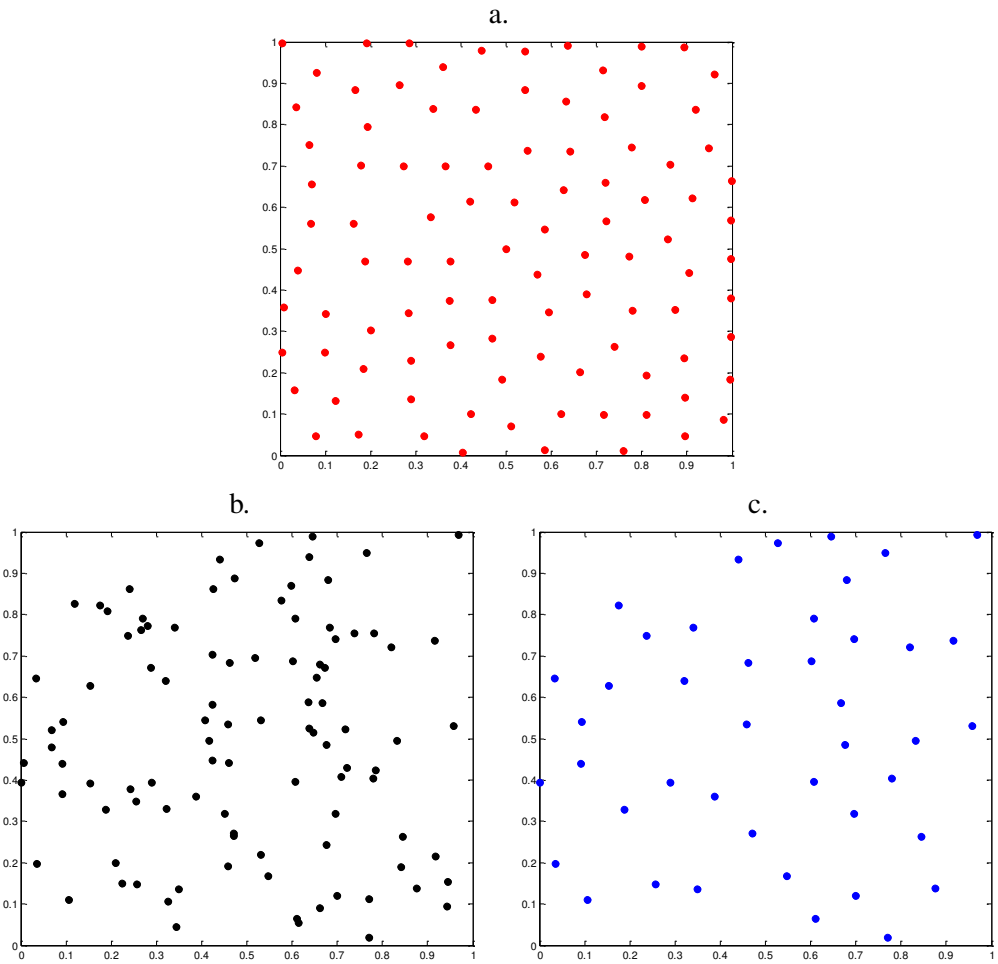


FIGURE 1.53 – Réparation du plan en deux dimensions et 100 points contenant des amas et des lacunes. **a)** Plan de référence, **b)** plan initial, **c)** plan sans amas à 43 points obtenu pour $d_{min} = 0.094$.

Tableau 1.3 – Valeurs *Mindist* et *Coverage* pour le plan de départ avant et après suppression des amas.

	Nombre de points	***Mindist***	***Coverage***
Plan de référence	100	0.094	0.015
Plan initial	100	0.006	0.580
Plan après suppression des amas	43	0.094	0.176

Nous pouvons observer que l'élimination des 57 points, qui sont donc considérés comme trop proches d'un autre point, entraine une amélioration en termes d'uniformité caractérisée par une augmentation de la valeur *Mindist* (de 0.006 à 0.094) et une diminution de la valeur *Coverage* (de 0.580 à 0.176). Pour synthétiser tout cela nous pouvons aussi représenter la distribution des valeurs des distances minimales sous la forme de graphes "box plot" [61, 62] (figure 1.54) :

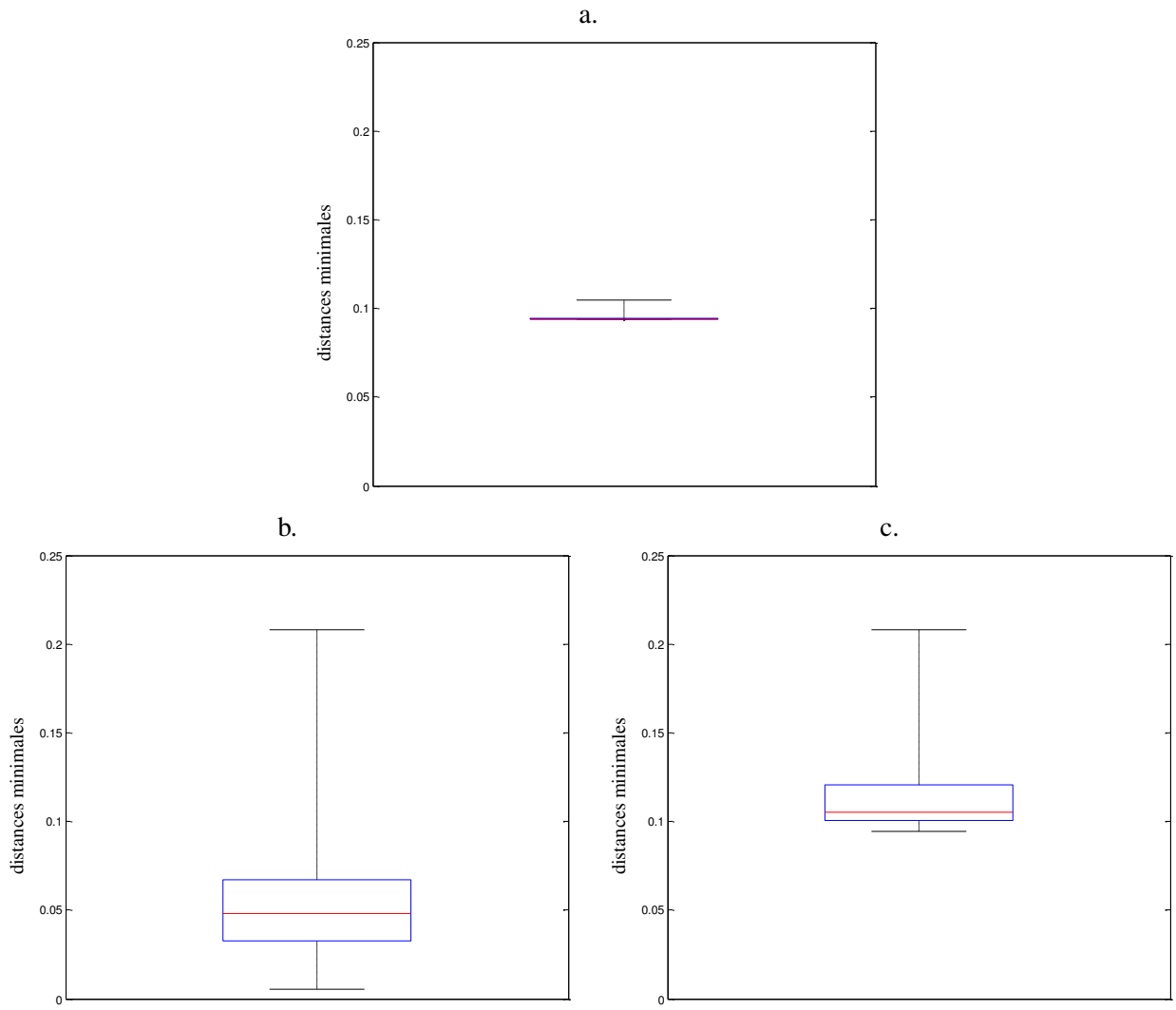


FIGURE 1.54 – Box plots des distances minimales **a)** du plan de référence, **b)** du plan initial contenant des amas et des lacunes, **c)** du plan après élimination des amas.

Les figures 1.54 a. et b. représentent respectivement la distribution des distances minimales pour le plan de référence à 100 points et le plan initial avec amas et lacunes alors que la figure 1.54 c.

correspond au plan après élimination des amas. Nous savons que pour un plan d'expériences uniforme les distances entre les points sont homogènes ce qui se traduit par un intervalle interquantile très réduit donc une superposition des valeurs minimales et maximales sur le graphe box plot (figure 1.54 a.), contrairement au box plot du plan mal conditionné (figure 1.54 b.) qui montre une forte dispersion des distances minimales, caractérisant la présence d'amas et de lacunes. La représentation des distances minimales du plan après élimination des amas, montre une augmentation des valeurs, avec une valeur minimale égale à celle du plan de référence. Toutefois, l'étendue demeure grande ce qui s'explique par la présence de lacunes.

4.2 Remplissage des lacunes

L'autre étape de réparation consiste à combler les zones lacunaires qui, contrairement aux amas, sont plus pénalisantes car elles sont représentatives d'un manque d'information dans certaines zones de l'espace.

Quand cela est possible, nous proposons d'ajouter des points pour combler ces lacunes à l'aide de l'algorithme WSP. Mais cet algorithme est un algorithme de sélection qui permet uniquement de choisir un ensemble de points parmi un ensemble de points candidats et ne permet donc pas d'ajouter des points. Pour pallier cette insuffisance, nous proposons de concaténer deux plans d'expériences : le plan lacunaire constitué de points appelés "points protégés" auquel nous ajoutons un deuxième plan uniforme contenant un très grand nombre de points afin de couvrir tout le domaine, l'ensemble constituant les "points candidats". Par le terme "points protégés" nous entendons que ces points ne seront jamais éliminés par l'algorithme et qu'ils seront obligatoirement dans le plan final.

Avant de combler les lacunes, nous effectuons une première étape qui consiste à supprimer tous les points candidats situés à une distance inférieure à la valeur d_{min} de référence d'un point protégé. Ainsi, par cette première étape nous nous affranchissons de la possibilité de sélectionner des points à proximité des points protégés. La deuxième étape va consister à appliquer l'algorithme WSP sur le plan global, constitué des points protégés et des points candidats restants après l'étape préliminaire, en fixant la valeur d_{min} du plan de référence : l'algorithme va ainsi progressivement combler les zones lacunaires tout en conservant les points protégés. L'algorithme de remplissage des lacunes est résumé ci-dessous (Algorithme 4.2).

Algorithme 4.2 Algorithme de remplissage de lacunes

Considérer un ensemble de points en D dimensions appelé points protégés, notés P
Considérer un ensemble de points candidats, notés C
Concaténer les deux ensembles de points
Fixer une valeur d_{min}
Éliminer de l'ensemble candidat tous les points C tels que $d_{PC} < d_{min}$, avec :

$$d_{PC} = \|x_P - x_C\| = \sqrt{\sum_{r=1}^{D} (x_{Pr} - x_{Cr})^2} = \text{distance entre les points } P \text{ et } C$$

Choisir un point protégé comme point initial de l'algorithme WSP, noté O
JUSQU'À ce que tous les points candidats soient sélectionnés ou éliminés

- Éliminer de l'ensemble candidat tous les points tels que $d_{OC} < d_{min}$
- Remplacer le point O par le point le plus proche parmi les points restants

FIN

Nous proposons d'appliquer cette étape de réparation sur l'exemple en deux dimensions présenté figure 1.51.

La figure 1.55 a. représente le plan initial (points bleus) auquel nous avons ajouté 5000 points candidats représentés par des ronds noirs. Nous pouvons observer que les points candidats couvrent tout l'espace des variables ce qui permet de proposer des points dans tout le domaine pour combler les zones lacunaires. Sur la figure 1.55 b., seuls les points candidats situés à une distance supérieure à la valeur d_{min} sont représentés, avec $d_{min} = 0.094$. C'est alors à partir de ces deux ensembles de points que nous appliquons l'algorithme WSP en protégeant les points du plan de départ.

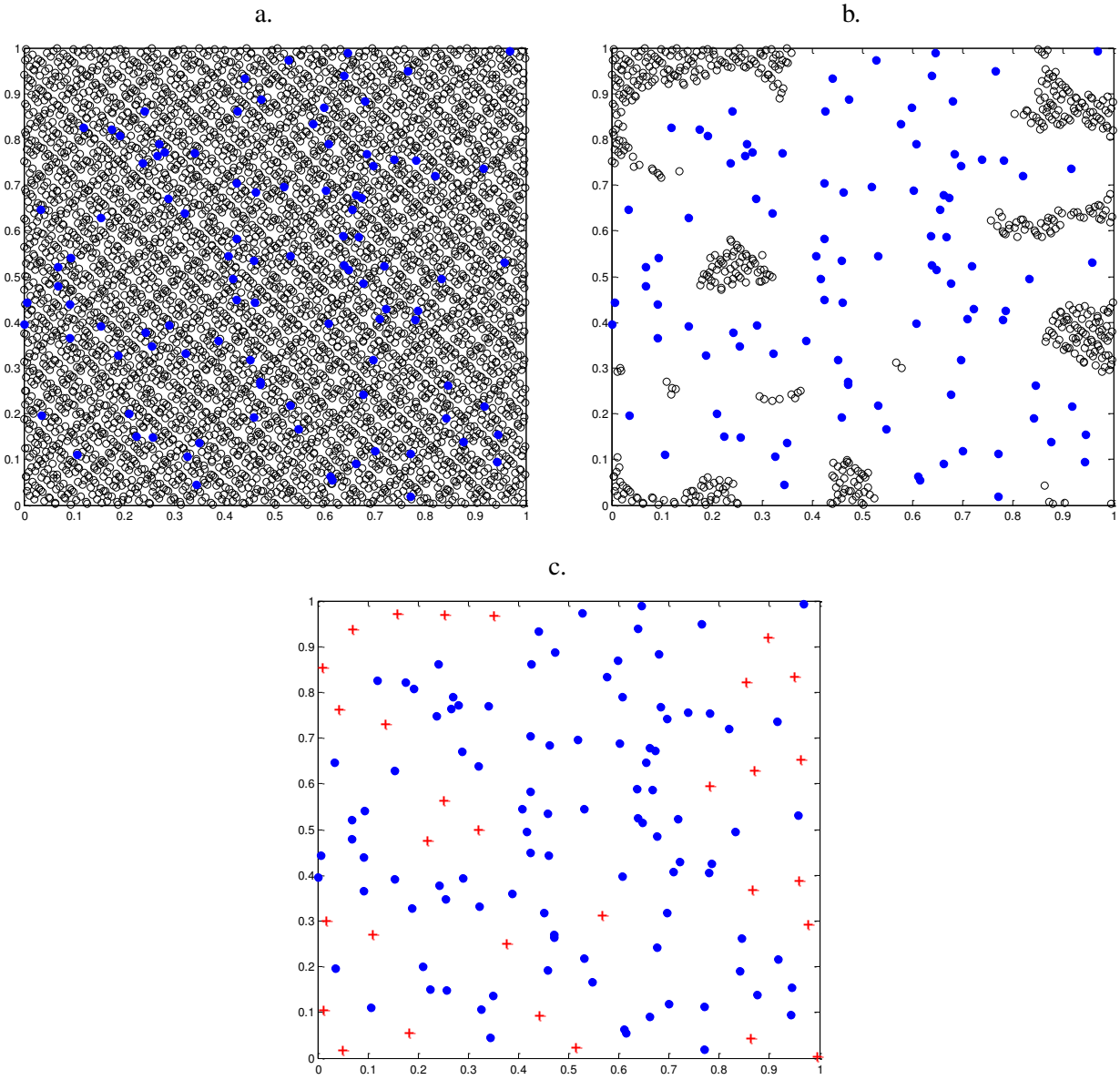


FIGURE 1.55 – Remplissage des lacunes d'un plan en deux dimensions et 100 points. **a)** Représentation du plan (**100** points) concaténé à un ensemble de 5000 points candidats. **b)** Représentation du plan avec les points candidats restants situés à une distance supérieure à la valeur d_{min} d'un point protégé. **c)** Plan sans lacunes (**130** points), les points ajoutés sont représentés par des croix rouges.

L'étape de comblement des lacunes conduit à l'ajout de 30 points qui sont représentés par des croix rouges sur la figure 1.55 c. Le plan sans lacunes compte alors 130 points et présente une valeur *Mindist* inchangée ($Mindist = 0.006$), une légère amélioration de la valeur *MoyMin* (de 0.053 à 0.061) et une valeur *Coverage* diminuée passant de 0.580 pour le plan de départ à 0.479 pour le plan sans lacunes.

Nous proposons de compléter l'interprétation des critères par la figure 1.56 qui représente le box plot des distances minimales du plan sans lacunes.

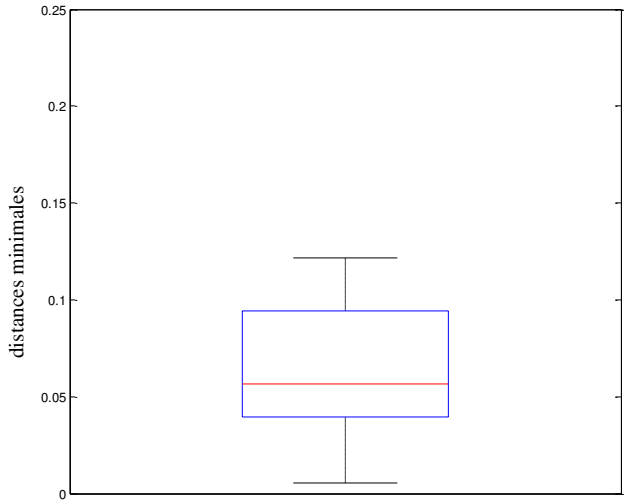


FIGURE 1.56 – Box plot des distances minimales du plan à **130** points après l'étape de remplissage des lacunes.

Sur la figure 1.56 nous observons que l'ajout des 30 points au plan de départ conduit à une diminution de l'étendue des distances minimales avec une valeur minimale qui reste égale à la valeur du plan initial ($= 0.006$) et la valeur maximale qui diminue de 0.208 à 0.121. Ces critères intrinsèques sont regroupés dans le tableau 1.4.

Tableau 1.4 – Tableau regroupant les valeurs des critères intrinsèques du plan en 2D avec 100 points avant et après l'étape de remplissage des lacunes.

	Nombre de points	***Mindist***	***Coverage***
Plan de référence	100	0.094	0.015
Plan initial	100	0.006	0.580
Plan sans lacunes	130	0.006	0.479

4.3 Stratégies de reconditionnement

Lorsqu'un ensemble de points est mal conditionné, nous avons proposé deux méthodes de réparation à savoir la suppression des amas et le remplissage des lacunes. Nous pouvons alors nous demander si ces deux méthodes doivent être utilisée indépendamment ou si elles doivent être appliquées successivement, et dans ce cas dans quel ordre. Le choix de la stratégie dépendra des contraintes expérimentales et de l'objectif. Aussi, nous proposons les quatre stratégies suivantes :

— Stratégie 1 : pour réparer un plan auquel nous ne pouvons pas ajouter de points, nous n'envisagerons que l'étape d'**élimination des amas**, si nécessaire Par exemple, cette démarche pourra être utilisée pour sélectionner des points pour une étape de modélisation,les points restants servant à tester la validité du modèle,

— Stratégie 2 : dans le cas où nous nous intéressons qu'aux zones du domaine déficientes en information, nous n'envisagerons que l'étape de **remplissage des lacunes**,

— Stratégie 3 : dans un premier temps, nous appliquerons l'étape d'**élimination des amas** puis l'étape de **comblement des lacunes**,

— Stratégie 4 : nous commencerons par **combler les lacunes** puis nous **supprimerons les amas**.

Afin d'illustrer et de comparer ces quatre stratégies, nous proposons de les appliquer sur le plan de départ de la figure 1.51, qui présente des amas et des lacunes. Nous rappelons que la matrice de référence en 2D et 100 points présente une valeur *Mindist* égale à 0.094 ce qui détermine la valeur d_{min} des algorithmes d'élimination des amas et de remplissage des lacunes quelle que soit la stratégie envisagée. Les répartitions des points de ces quatre plans sont représentées sur les figures 1.57 à 1.60 et leurs critères intrinsèques sont reportés dans le tableau 1.5.

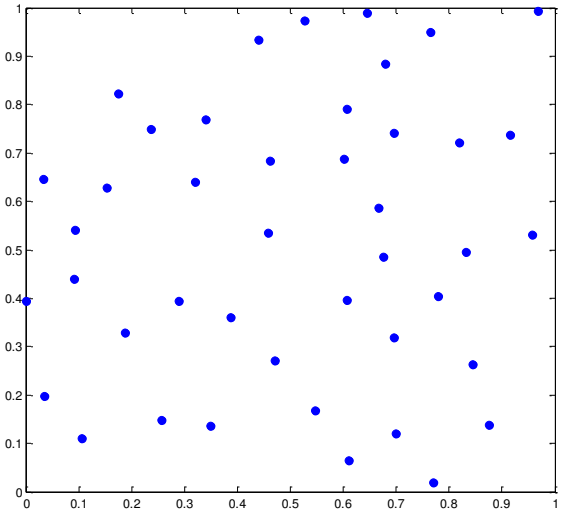


FIGURE 1.57 – Première stratégie de reconditionnement : suppression des amas, qui conduit à un plan sans amas à **43** points.

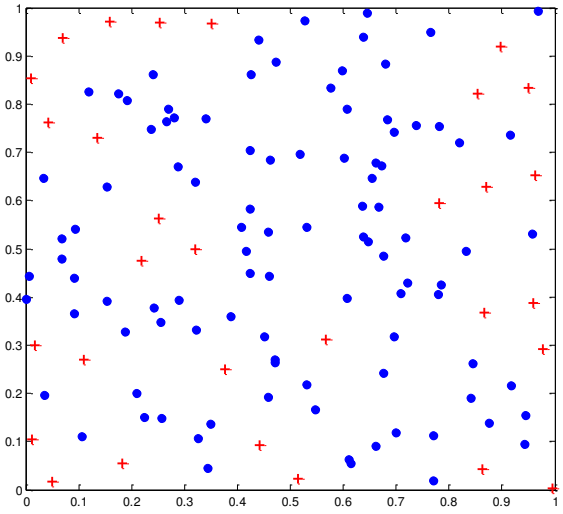


FIGURE 1.58 – Deuxième stratégie de reconditionnement : remplissage des lacunes, qui conduit à un plan sans lacunes à **130** points.

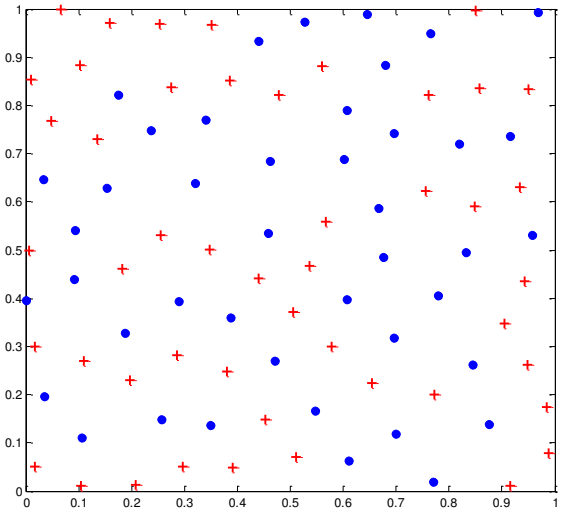


FIGURE 1.59 – Troisième stratégie de reconditionnement : suppression des amas puis remplissage des lacunes, qui conduit à un plan sans amas et sans lacunes à **91** points.

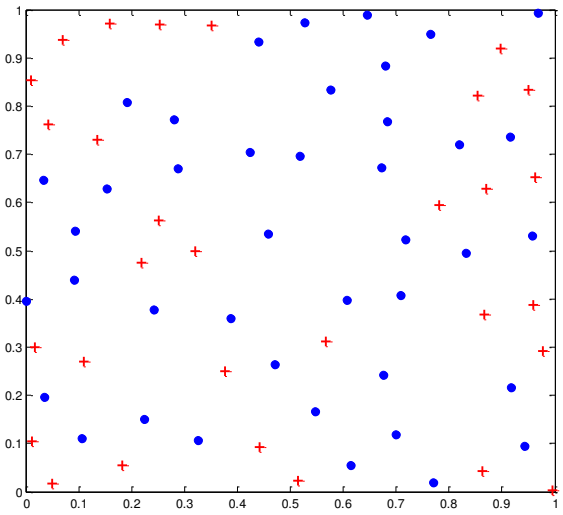


FIGURE 1.60 – Quatrième stratégie de reconditionnement : remplissage des lacunes puis suppression des amas, qui conduit à un plan sans amas et sans lacunes à **71** points.

Nous proposons de compléter la représentation graphique des plans par la figure 1.61 qui représente les box plots des distances minimales des plans issus des stratégies 3 et 4, ceux des stratgéies 1 et 2 sont représentés respectivement par les figures 1.54 c. et 1.56.

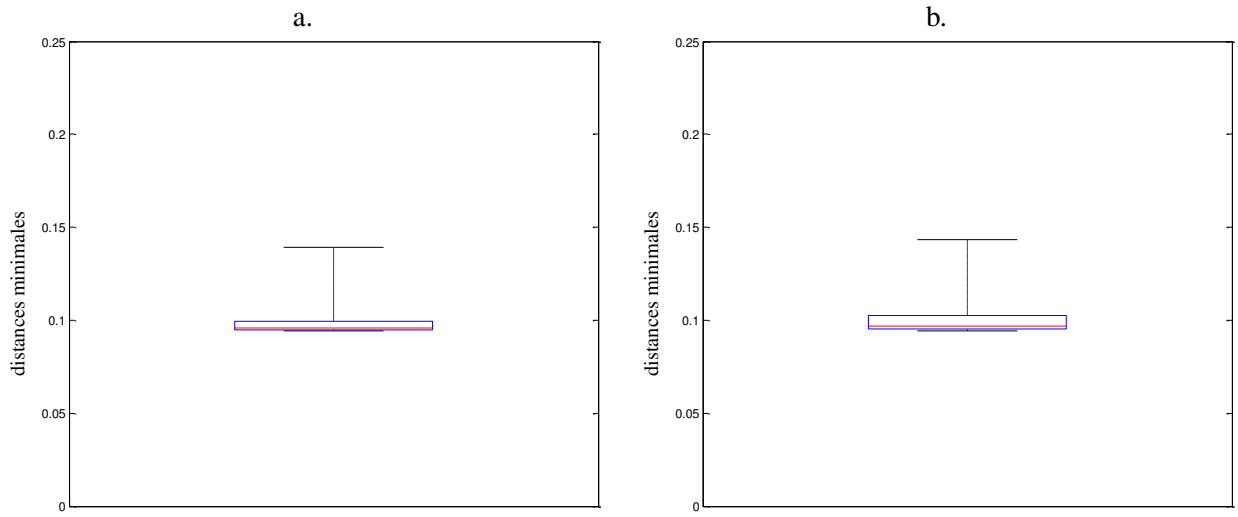


FIGURE 1.61 – Box plot des distances minimales, **a)** du plan à **91** points issu de la stratégie 3 et **b)** du plan à **71** points obtenu par la stratégie 4.

Les figures 1.61 a. et b. représentent respectivement les distances minimales pour le plan à 91 points obtenu pour la troisième stratégie et le plan à 71 points obtenu par la quatrième stratégie. Nous observons que pour ces deux stratégies la valeur minimale est augmentée passant de 0.006 pour le plan initial à 0.094 ce qui s'explique par la suppression des amas, alors que la valeur maximale est diminuée passant de 0.208 pour le plan initial à 0.139 pour le plan de la stratégie 3 et 0.144 pour le plan de la stratégie 4, ce qui est la conséquence du remplissage des lacunes.

Pour les plans issus des quatre stratégies, nous regroupons dans le tableau 1.5 les valeurs des critères intrinsèques et le nombre de points.

Tableau 1.5 – Tableau des critères intrinsèques des différents plans envisagés dans cette étude.

	Nombre de points	***Mindist***	***MoyMin***	***Coverage***	***écart-type***
Plan de référence	100	0.094	0.094	0.015	0.014
Plan initial	100	0.006	0.052	0.580	0.031
Stratégie 1	43	0.095	0.112	0.176	0.020
Stratégie 2	130	0.006	0.001	0.479	0.029
Stratégie 3	91	0.094	0.099	0.076	0.008
Stratégie 4	71	0.094	0.101	0.089	0.009

A partir du tableau 1.5, nous pouvons commenter les valeurs des critères intrinsèques des plans obtenus par l'application successive des deux étapes de réparation :

— Stratégie 3 : la première étape, l'élimination des amas, conduit à éliminer 57 points et à n'en conserver que 43. Puis, le comblement des lacunes en imposant les 43 points protégés ajoute 48 points et permet d'obtenir une répartition de 91 points uniforme, caractérisée par une valeur *Mindist* égale à celle du plan de référence (= 0.094) et une faible valeur *Coverage* (= 0.076),

— Stratégie 4 : la première étape, le comblement des lacunes, ajoute 30 points. La seconde étape de suppression des amas en utilisant toujours la même valeur d_{min} conduit à un ensemble de 71 points. Cette répartition de points uniforme est caractérisée par une valeur *Mindist* égale à celle du plan de référence ($= 0.094$) et une faible valeur *Coverage* ($= 0.089$).

Quel que soit l'ordre de ces deux étapes de réparation, nous obtenons une amélioration des critères intrinsèques (tableau 1.5) et une répartition des points dans l'espace qui est plus uniforme (figures 1.59 et 1.60). Toutefois, même si la répartition finale des points est plus uniforme, ces deux stratégies présentent un inconvénient. En effet, la troisième stratégie débute par la suppression des amas puis ajoute des points pour remplir les zones déficientes en information ce qui peut conduire à l'ajout de nouveaux points proches de ceux préalablement supprimés. C'est pourquoi cette stratégie s'avère coûteuse en termes de nombre d'expériences. La figure 1.62 montre le plan obtenu par cette stratégie en ajoutant les points qui ont été considérés comme des amas et qui ont donc été supprimés à l'étape 1, ces points sont représentés par des carrés noirs.

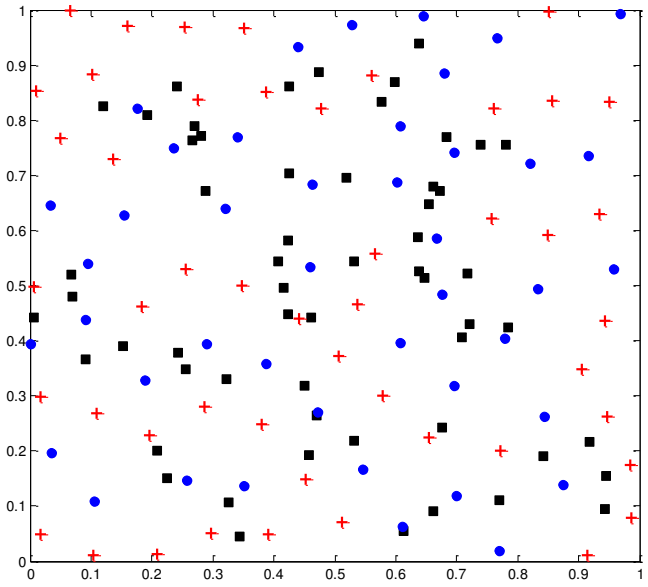


FIGURE 1.62 – Plan obtenu par la stratégie 3 auquel nous avons ajouté les points qui ont été supprimés lors de la première étape. Ces points sont représentés par des carrés noirs.

Sur la figure 1.62, nous pouvons observer que certaines croix rouges qui correspondent aux points ajoutés lors de la deuxième étape sont proches des carrés noirs et donc des points supprimés à la première étape.

La stratégie 4, qui débute par le remplissage des lacunes, permet d'éviter l'ajout de points proches de ceux supprimés ce qui rend cette stratégie moins coûteuse en nombre d'expériences que la stratégie 3. Cependant, la deuxième étape de suppression des amas peut conduire à la création de nouvelles zones lacunaires. Sur la figure 1.63, nous mettons en évidence ces nouvelles zones lacunaires, représentées par des cercles noirs.

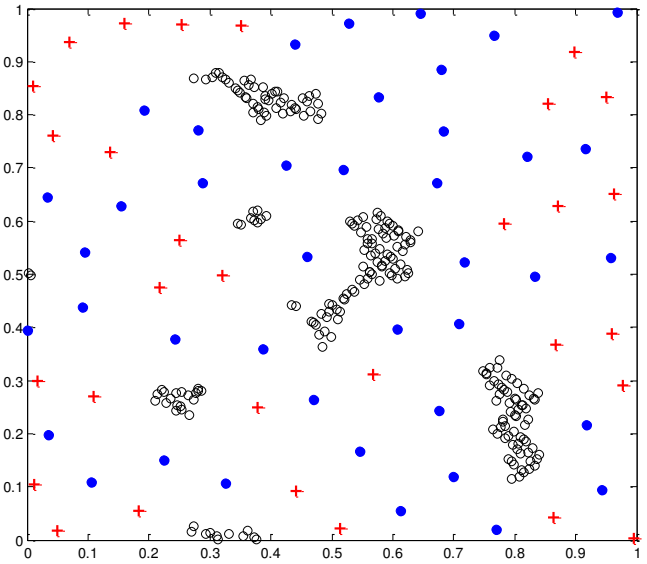


FIGURE 1.63 – Plan obtenu par la stratégie 4 à partir duquel nous mettons en évidence la présence de zones lacunaires.

Sur la figure 1.63, nous observons la présence de six zones lacunaires ce qui signifie que par cette stratégie, même si nous améliorons l'uniformité de la répartition des points, nous n'avons pas combler toutes les zones lacunaires initiales.

Les plans issus des différentes stratégies de reconditionnement ne présentant pas tous le même nombre de points, nous proposons d'utiliser comme critère l'*écart-type* qui est indépendant du nombre de points. La figure 1.64 permet de comparer ces valeurs pour chaque plan résultant des différentes stratégies.

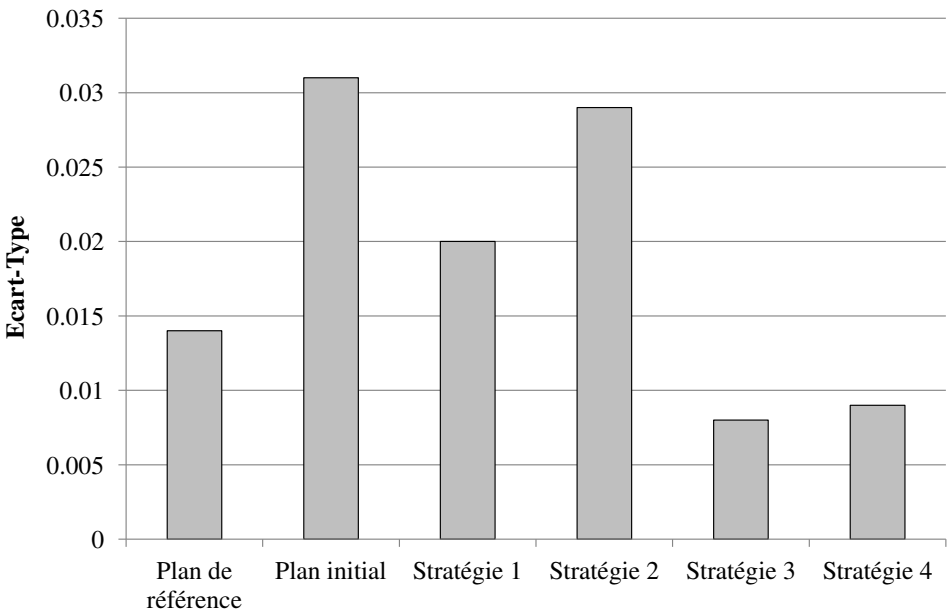


FIGURE 1.64 – Représentation des valeurs de l'*écart-type* pour le plan de référence, le plan initial et les 4 plans issus des différentes stratégies de reconditionnement.

La figure 1.64, montre une forte différence des valeurs de l'*écart-type* selon la stratégie considérée. Nous observons que les valeurs les plus élevées sont obtenues pour le plan initial et le plan issu de la deuxième stratégie à savoir le remplissage des lacunes, ce qui s'explique par la présence de points très proches. *A contrario*, la première stratégie qui consiste à éliminer les amas permet de diminuer la valeur de l'*écart-type*, mais les valeurs les plus faibles sont obtenues pour les stratégies qui utilisent les deux méthodes de reconditionnement.

La comparaison de ces quatre stratégies nous permet de conclure que les plans les mieux "réparés" résultent de l'utilisation successive de la méthode d'élimination des amas et de remplissage des lacunes, quel que soit l'ordre, et s'accompagne d'une amélioration des critères intrinsèques.

4.4 Conclusion sur le reconditionnement

Nous savons qu'une mauvaise répartition des points dans l'espace des variables, caractérisée par une accumulation de points en certaines zones et/ou des zones vides, peut être pénalisante pour des études de surfaces de réponse. Aussi, nous avons proposé d'utiliser l'algorithme WSP pour éliminer les amas si nécessaire et combler les zones lacunaires, car il a l'avantage d'être facile et rapide à mettre en œuvre même en grande dimension. Les résultats obtenus à partir de l'exemple en deux dimensions sont prometteurs puisque nous avons montré que l'utilisation consécutive de ces deux étapes conduit à des structures présentant de meilleurs critères intrinsèques, c'est à dire des distributions de points plus uniformes. Seules les contraintes expérimentales orienteront le choix de l'ordre des étapes.

Dans cet exemple en deux dimensions, nous avons pu visualiser les répartitions de points dans l'espace des variables mais en grande dimension il faudra débuter l'étude par une Analyse en Composantes Curvilignes pour repérer la présence éventuelle d'amas dans l'espace multidimensionnel initial.

Conclusion

Dans cette première partie, les méthodes de visualisation de données et de sélection de points ou de variables ont été présentées de manière théorique en tant qu'outils, mais aussi avec les améliorations que nous avons jugées nécessaires pour répondre aux problématiques de la grande dimension. Dans ce "catalogue" de méthodes, nous avons sélectionné les plus appropriées, à savoir :

- l'Analyse en Composantes Curvilignes pour la visualisation des données en grande dimension avec la possibilité de détecter les amas de points,
- l'algorithme WSP pour sélectionner des sous-ensembles de points ou réparer des plans mal conditionnés en éliminant les amas et en remplissant les lacunes,
- l'algorithme WSP modifié pour :
  - densifier une zone d'intérêt, par l'algorithme aWSP,
  - réduire le nombre de variables, par l'algorithme V-WSP.

Dans la deuxième partie de ce manuscrit, nous proposerons d'appliquer ces outils dans divers domaines à savoir les études de relations structure-activité (QSAR), les données spectroscopiques et la simulation numérique en utilisant une ou plusieurs méthodes pré-citées.

Deuxième partie
Cas d’étude

Introduction

Dans la première partie du manuscrit, nous avons proposé des méthodes permettant de visualiser les données, de réduire la dimension par une sélection représentative de la population initiale et de ”réparer” les structures lorsque les données ne sont pas bien réparties dans le domaine. Parmi toutes ces méthodes, nous avons proposé de ne retenir que les plus pertinentes afin de constituer un ”catalogue” qui permettra de choisir la ”bonne” méthode en fonction du type de données et de l'objectif de l'étude. Cette seconde partie sera consacrée à des cas d'étude dans trois domaines d'application, en mettant en évidence pour chacun d'eux, les avantages et les inconvénients des différentes approches. Ainsi, en fonction des objectifs et des contraintes, nous choisirons dans le catalogue des méthodes, la ou les approches les plus pertinentes.

Le **premier chapitre** sera consacré aux études des relations structure-activité (QSAR) qui mettent souvent en jeu un très grand nombre de descripteurs. Le traitement de la grande dimension réside dans ces études dans la gestion de tableau présentant un grand nombre de colonnes (plusieurs centaines) et un nombre moindre de lignes. Nous proposons alors de visualiser les données par l'Analyse en Composantes Curvilignes et/ou de réaliser une sélection des descripteurs les plus représentatifs par l'algorithme V-WSP pour construire un modèle prévisionnel.

Dans le **second chapitre**, nous présenterons des applications à partir de données spectroscopiques. Dans ce domaine, il est courant de travailler avec des tableaux présentant un grand nombre de lignes et de colonnes. Il nous a paru pertinent dans ce contexte de comparer les algorithmes de sélection de points pour la construction de sous-ensembles de calibration et de validation.

Enfin, dans le **troisième chapitre**, nous nous intéresserons au domaine de la simulation numérique et plus particulièrement aux méthodes de reconditionnement afin de ”réparer” les plans d'expériences pour répartir au mieux les points dans l'espace des variables. Nous distinguerons le cas où la ”réparation” est réalisée *a priori* de la ”réparation” d'un plan d'expériences après projection (ou repliage) des points dans un sous-ensemble de variables considérées comme influentes.

Chapitre 1

Les études QSAR

Depuis longtemps, les chimistes cherchent à établir des liens quantitatifs entre des variables de sortie (activité biologique, toxicité, écotoxicologie, affinité pour un récepteur, ...), et la nature de molécules sensées être responsables de la variation de ces réponses. Comme il n'est pas possible d'établir un modèle en fonction de variables qualitatives, on utilisera des descripteurs qui décriront quantitativement les propriétés moléculaires. Ainsi, les études QSAR (**Q**uantitative **S**tructure-**A**ctivity **R**elationship) ont été développées pour établir une relation mathématique entre des réponses expérimentales et des descripteurs, pour une série de composés chimiques similaires [63].

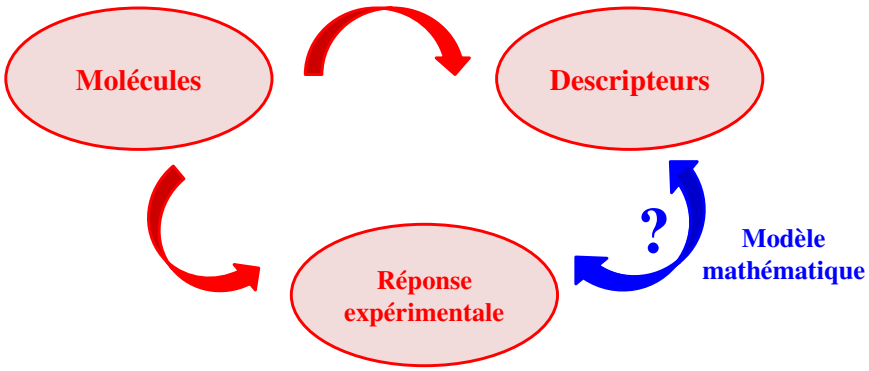


FIGURE 2.1 – Principe des études QSAR.

La première étape des méthodes QSAR consiste à choisir des indicateurs qui décrivent la structure et caractérisent les proriétés physico-chimiques des molécules.

Les descripteurs peuvent être regroupés par type, parmi lesquels on trouve :

- les descripteurs constitutionnels [64] qui reposent sur la structure chimique de la molécule comme la composition élémentaire, les groupements fonctionnels, ...
- les descripteurs topologiques [65] qui sont obtenus à partir de la structure bi-dimensionnelle,
- les descripteurs géométriques qui sont évalués à partir de la structure tri-dimensionnelle comme les distances, les angles, les angles dièdres, ...
- les descripteurs quantiques [66, 67, 68, 69] qui sont issus de la structure électronique de la molécule comme la distribution de charge, les énergies HOMO/LUMO, l'électronégativité, ...

- les descripteurs énergétiques comme l'énergie de dissociation, l'énergie d'atomisation, ...
- les descripteurs empiriques comme la masse moléculaire, logP, ...

Les études QSAR requièrent différentes étapes : la construction du tableau de données, la sélection d'un sous-ensemble de molécules représentatif, le calcul du modèle et sa validation. Le choix de la base de données est décisif car elle doit contenir des descripteurs pertinents et le moins corrélés possible. Pour établir le "bon" modèle, il peut être utile de sélectionner au préalable les descripteurs les plus orthogonaux puis des algorithmes de sélection de points permettront de retenir le sous-ensemble de molécules de qualité optimale pour estimer les coefficients du modèle. Une fois le modèle construit, il sera ensuite validé avant d'être utilisé pour prévoir les propriétés physicochimiques et les activités d'autres molécules ou pour concevoir de nouvelles structures.

Il apparait clairement que le choix des descripteurs est l'étape critique des études QSAR, car ils doivent permettre de discriminer les molécules en fonction de leur comportement. Une première réponse pourrait être apportée par la position des différentes molécules dans l'espace de ces descripteurs. En effet, on peut s'attendre à ce que deux points proches présentent des résultats similaires. Il faut donc s'interroger *a priori* sur la qualité de la structure que représente l'ensemble des molécules dans l'espace des variables d'entrée et pour cela on pourrait faire appel à l'Analyse en Composantes Curvilignes comme méthode de visualisation. Dans le cas où les descripteurs sont en grand nombre, il sera judicieux d'en sélectionner un sous-ensemble constitué des moins corrélés tout comme, dans le cas d'un grand nombre de molécules, il faudra sélectionner les plus représentatives.

Nous présenterons et traiterons trois cas d'étude, issus de différents domaines d'application et pour chacun d'eux, nous proposerons d'utiliser l'une des méthodes du "catalogue" présenté dans la Partie 1 pour répondre au mieux aux objectifs.

1.1 Cas d'étude N°1 : préparation d'échantillons MALDI

Les données présentées sont issues du projet **DESIRS** (**DE**sign of **S**olid state experiments for **I**onization of copolyme**R**s in mass **S**pectrometry) dans le cadre du programme "jeunes chercheuses et jeunes chercheurs" financé par l'Agence Nationale de la Recherche (ANR).

Le développement de techniques d'ionisation douce, telles que la désorption/ionisation laser assistée par matrice (MALDI), a permis la production d'ions intacts de haute masse molaire en phase gazeuse. De ce fait, la spectrométrie de masse a révolutionné le domaine de la caractérisation des polymères synthétiques [70, 71]. La spectrométrie de masse est devenue une technique très précieuse pour l'analyse de polymères car une seule mesure révèle presque autant d'informations que l'ensemble des techniques de caractérisation traditionnelles réunies. La technique MALDI se compare avantageusement à d'autres techniques mises en œuvre pour la détermination des masses moléculaires en raison de sa sensibilité et de sa rapidité, mais aussi parce qu'elle n'est pas confrontée aux mêmes limitations. Le spectre MALDI permet d'atteindre facilement la masse moléculaire moyenne en nombre (M_n) et la masse moléculaire moyenne en masse (M_w) ; l'indice de polydispersité (M_w/M_n) peut donc également être calculé à partir des données spectrales obtenues. Toutefois, un problème majeur qui limite les applications généralisées de la technique MALDI est la mauvaise connaissance des processus fondamentaux qui régissent la désorption et l'ionisation des analytes. En conséquence, le développement des méthodes de préparation des échantillons MALDI reste assez empirique et devient extrêmement difficile quand il s'agit de systèmes polymériques complexes tels que les copolymères à blocs amphiphiles. En effet, un point-clé de la réussite d'une analyse MALDI-MS est la préparation des échantillons, avec notamment la sélection appropriée de la matrice et de l'agent d'ionisation. En raison de la diversité des matériaux polymères et du rôle ambigu de la matrice dans la désorption/ionisation de l'analyte, aucun protocole standard n'est disponible. Même au sein d'une classe de polymères, la préparation de la masse moléculaire des échantillons doit être adaptée en fonction de l'analyte et la taille de l'agent cationique ainsi que la proportion de matrice doivent être optimisées en conséquence [72].

Pour pallier la complexité de la préparation des échantillons MALDI, une approche QSAR a été envisagée afin de relier les effets de la matrice et de l'agent d'ionisation aux performances de la méthode. L'étude cherche à identifier les paramètres influençant l'efficacité du processus d'ionisation en s'intéressant à un homopolymère de polyéthylène glycol (PEG) de petite taille ($2kD$). Pour cela, à partir des travaux antérieurs et de résultats apportés dans la littérature, 19 matrices et 26 sels ont été sélectionnés formant ainsi 494 couples sel-matrice candidats (tableau 2.1).

Tableau 2.1 – Matrices et agents de cationisation sélectionnés pour le MALDI des homopolymères.

Matrices	**Dith** : dithranol ou 1,8,9-anthracenetriol
	2,3-DHB : 2,3-dihydroxybenzoic acid
	2,4-DHB : 2,4-dihydroxybenzoic acid
	2,5-DHB : 2,5-dihydroxybenzoic acid
	2,6-DHB : 2,6-dihydroxybenzoic acid
	HABA : 2-(4'-hydroxybenzeneazo)benzoic acid
	THAP : 2',4',6'-trihydroxyacetophenone
	DHBQ : 2,5-dihydroxy-p-benzoquinone
	MBT : 2-mercaptobenzothiazole
	SA (Sinapinic acid) : 3,5-dimethoxy-4-hydroxycinnamic acid
	HPA : 3-hydroxypicolinic acid
	FA (Ferulic acid) : 3-methoxy-4-hydroxycinnamic acid
	CMBT : 5-chloro-2-mercaptobenzothiazole
	5-CSA : 5-chlorosalicylic acid
	9-ACA : anthracene-9-carbonic acid
	NOR (Nor-harmane) : 9H-pyrido[3,4-b]indole
	9-NA : 9-nitroanthracene
	HCCA : α- cyano- 4- hydroxycinnamic acid
	IAA : 3-indoleacrylic acid
Sels	Li-F ; Li-Cl ; Li-Br ; Li-I ;
	Na-F ; Na-Cl ; Na-Br ; Na-I ;
	K-F ; K-Cl ; K-Br ; K-I ;
	Rb-F ; Rb-Cl ; Rb-Br ; Rb-I ;
	$Ag - NO_3$; Ag-F ; Ag-Cl ; Ag-Br ; Ag-I ;
	Cu-Cl ; $Cu - Cl_2$; Cu-Br ; Cu-I ; $Cu(NO_3)_2$

Dans cette étude, les matrices et les sels ont été respectivement caractérisés par 7 et 3 descripteurs regroupés dans le tableau 2.2 :

Tableau 2.2 – Descripteurs caractérisant les couples sel-matrice.

Matrices	X1	Masse molaire ($g.mol^{-1}$)
	X2	pKa
	X3	Moment dipolaire (Debye)
	X4	Absorption (337 nm)
	X5	Fluorescence (337 nm)
	X6	Énergie d'ionisation (eV)
	X7	Affinité protonique ($kJ.mol^{-1}$)
Sels	X8	Rayon atomique
	X9	Rayon ionique
	X10	Énergie de dissociation

Nous proposons de débuter cette étude QSAR par une ACC sur la base candidate regroupant les 494 couples sel-matrice. Nous pourrons ainsi identifier, à partir des descripteurs considérés dans cette étude, les couples qui présentent des propriétés similaires et pour lesquels nous pouvons nous attendre à avoir un comportement proche lors de l'analyse MALDI ou ceux qui auront des propriétés antagonistes.

Sur la projection ACC représentée par la figure 2.2 nous avons ajouté un code couleur qui nous permet d'identifier les couples en fonction du cation utilisé pour former le sel. Les 26 sels étudiés sont obtenus à partir de 6 cations : le litium (Li), le sodium (Na), le potassium (K), le rubidium (Rb), l'argent (Ag) et le cuivre (Cu).

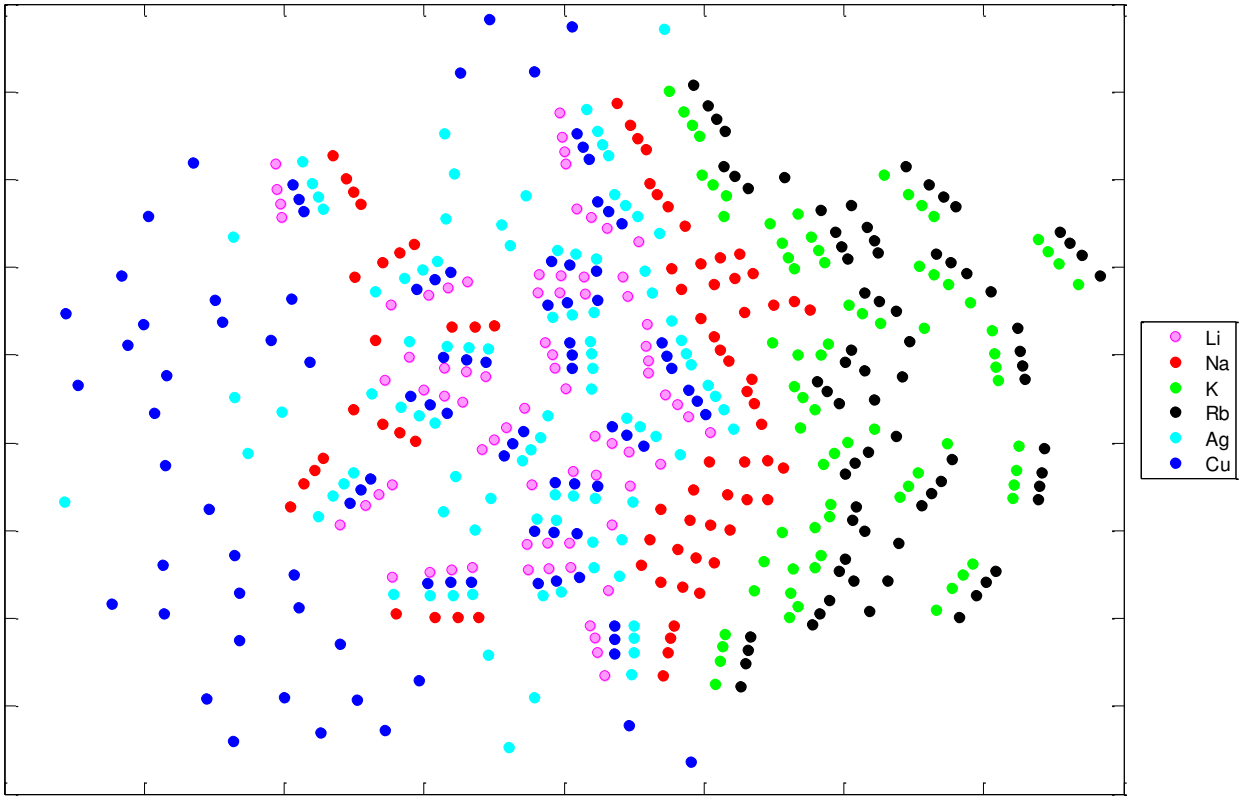


FIGURE 2.2 – Visualisation des couples sel-matrice par ACC avec un code couleur caractérisant les 6 cations formant les 26 sels des couples sel-matrice.

A partir de cette projection ACC, nous observons que les couples sel-matrice contenant les cations K et Rb (respectivement représentés en vert et noir) sont tous regroupés sur la partie droite de la projection ACC ce qui signifie que ces couples ont des caractéristiques analogues mais aussi antagonistes aux couples retrouvés dans la partie gauche de la projection à savoir ceux formés par les cations Li, Ag et Cu. En effet, la frontière verticale formée par le cation Na sépare d'une part tous les couples contenant les cations K et Rb et d'autre part les couples avec les cations Li, Ag, Cu et Na. Nous constatons également que certains couples sont regroupés et présentent des alignements selon le cation formant le sel. Le plus souvent ces alignements comptent 4 points qui correspondent aux 4 anions associés au cation. Comme nous visualisons des couples sel-matrice, nous nous sommes alors demandé si ces regroupements pouvaient correspondre aux différentes matrices. Pour répondre à cette question, nous avons adapté le code couleur de la projection ACC que nous représentons par la figure 2.3. Dans ce cas, une matrice sera décrite par une couleur et un symbole.

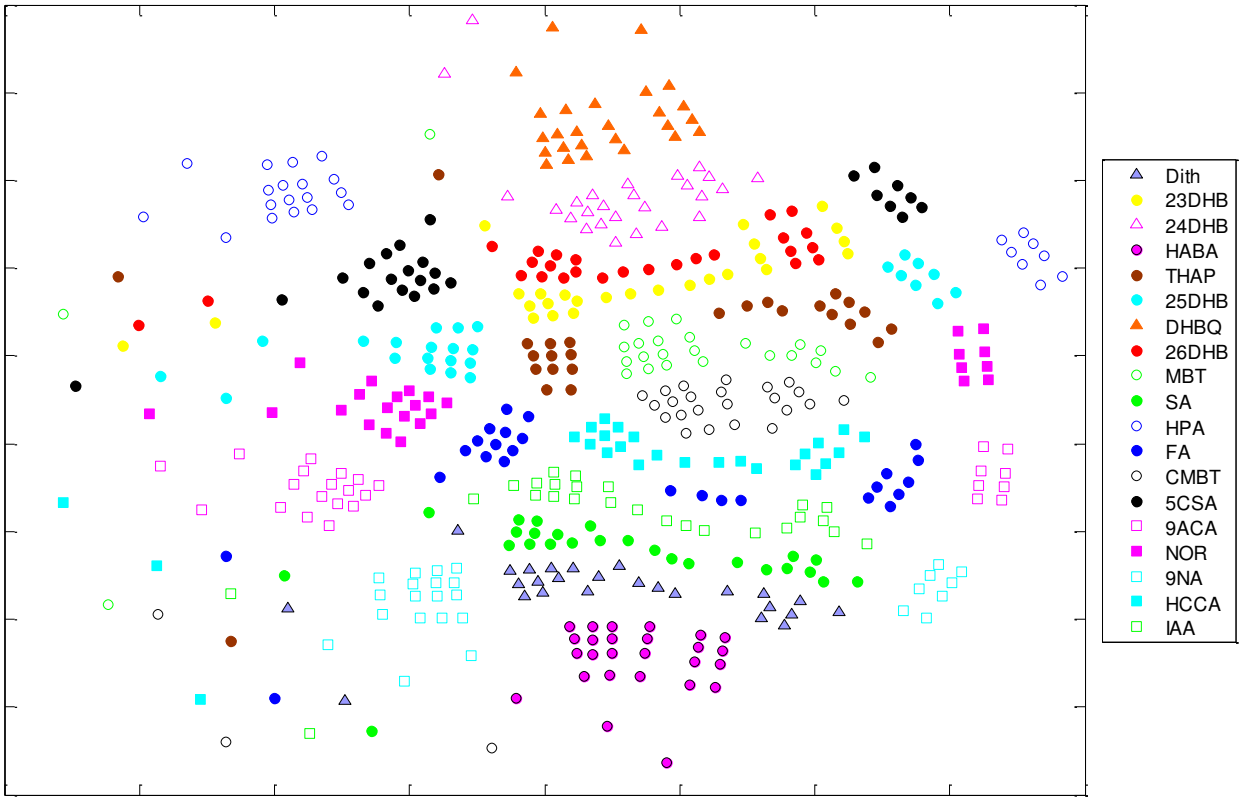


FIGURE 2.3 – Visualisation des couples sel-matrice par ACC avec un code couleur caractérisant les 19 matrices constituant les couples sel-matrice.

Si nous examinons la répartition des couples sel-matrice en fonction des 19 matrices étudiées, nous ne retrouvons pas les mêmes informations que celles obtenues à partir des sels. Sur la figure 2.3 nous observons que le comportement des couples sel-matrice dépend de la matrice utilisée. En effet, certains couples forment un seul groupe alors que d'autres sont séparés ou dispersés dans l'espace de projection. Nous constatons que les couples situés à proximité de la frontière verticale au centre du graphe (mise en évidence sur la figure 2.2), forment un seul groupe alors que ceux situés de part et d'autre de cette frontière sont séparés pour une même matrice. Ainsi, nous retrouvons sur la partie droite de la figure tous les couples contenant le K et le Rb alors qu'à gauche nous retrouvons les couples obtenus pour la même matrice mais avec les autres cations : Cu, Ag, Li et Na. Nous pouvons alors nous attendre à des caractéristiques différentes des matrices : HPA, 5-CSA, 2,5-DHB, THAP, NOR, FA, 9-ACA, 9-NA selon le sel utilisé avec peut-être des effets d'interaction et à des caractéristiques similaires quel que soit le sel pour les couples obtenus à partir des matrices : DHBQ ; 2,4-DHB ; 2,6-DHB ; 2,3-DHB ; MBT ; CMBT ; HCCA ; IAA ; SA ; Dith et HABA. Dans la partie inférieure gauche de la projection ACC, nous trouvons des couples isolés et obtenus par différentes matrices : MBT ; 5-CSA ; THAP ; 2,3-DHB ; 2,6-DHB ; HPA ; NOR ; 9-ACA ; 2,5-DHB ; HCCA ; CMBT ; FA ; IAA ; Dith ; SA ; 9-NA. Une analyse plus précise montre que ces couples utilisent le sel $Cu(NO_3)_2$ ce qui semble induire pour ce sel des caractéristiques particulières, et différentes des autres sels.

Comme nous l'avons dit en introduction, une étude QSAR débute par la construction d'une base de données candidate à partir de laquelle nous souhaitons choisir les couples sel-matrice les plus représentatifs. Dans cette étude, les 494 couples sont donc considérés comme candidats et nous

proposons l'algorithme WSP pour extraire un sous-ensemble à N points répartis le plus uniformément possible dans l'espace des variables sans postuler de modèle *a priori*. Cette sélection nous permettra d'envisager différents traitements (régression des moindres carrés, régression par machines à vecteurs supports (SVR), régression stepwise). Nous choisissons de retenir un sous-ensemble de $N = 25$ points (couples), qui présente une répartition des points la plus uniforme possible dans l'espace, tout en garantissant une bonne qualité de prévision (fonction de variance de la réponse prédite proche de 1).

Dans la démarche d'une étude QSAR, les 25 couples sel-matrice sélectionnés (regroupés dans le tableau 2.3) sont testés et pour chacun d'entre eux nous mesurons la "fluence" laser requise pour atteindre une intensité visée : une "fluence" faible sera favorable.

Tableau 2.3 – Présentation des 25 couples sel-matrice sélectionnés et les valeurs de la réponse "fluence" obtenue pour chaque couple. Les couples en rouge ne permettent pas d'obtenir un résultat contrairement aux couples en vert.

Matrices	**Sels**	**Fluence**
24DHB	Li-F	-
24DHB	Rb-Cl	-
24DHB	$Cu(NO_3)_2$	**41%**
HABA	Li-Br	**42%**
HABA	K-F	-
THAP	Na-Br	**43%**
25DHB	K-F	-
DHBQ	Na-I	**60%**
DHBQ	$Cu - Cl_2$	**44%**
SA	K-I	-
SA	$Cu(NO_3)_2$	**31%**
HPA	Li-F	-
HPA	Rb-Cl	-
HPA	$Ag - NO_3$	**40%**
FA	$Cu - Cl_2$	**27%**
CMBT	Li-F	**32%**
CMBT	Rb-F	-
CMBT	$Cu(NO_3)_2$	**22%**
5-CSA	Ag-I	**29%**
9-ACA	Rb-Cl	-
9-ACA	$Ag - NO_3$	**45%**
NOR	Cu-Cl	**42%**
9-NA	Rb-F	-
9-NA	Ag-I	**44%**
IAA	$Ag - NO_3$	**26%**

Les 10 expériences en rouge dans le tableau ci-dessus, ne donnent pas de résultats, c'est-à-dire que même avec une puissance du laser au maximum, aucun spectre n'est obtenu. Nous rappelons que l'objectif est d'obtenir une valeur de fluence la plus faible.

La première étude consiste à réaliser une ACC en affectant une valeur de 100% à la réponse fluence pour les expériences qui n'ont pas donné de résultats (en rouge).

A partir de ces résultats, l'ACC va permettre de mettre en exergue les caractéristiques similaires entre les couples qui donnent de bons résultats et ceux donnant de mauvais résultats.

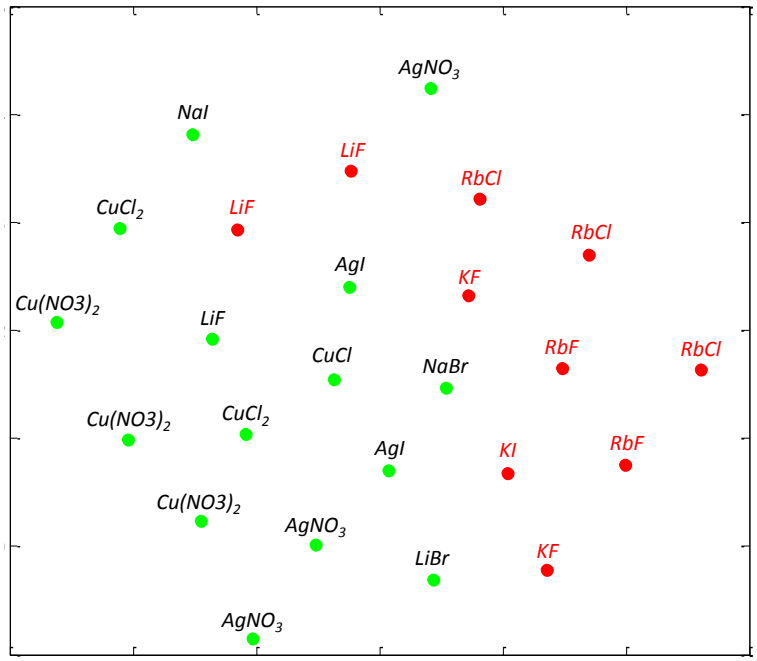


FIGURE 2.4 – Projection ACC de la solution à 25 couples.

Sur la figure ci-dessus, les 25 expériences sont séparées en deux groupes. Nous obtenons une séparation verticale avec d'une part les couples n'ayant pas donné de résultats (en rouge) et d'autre part ceux pour lesquels la fluence est inférieure à 60% (en vert). Il est important de préciser que les deux couples qui semblent mal classés et proches des couples qui fonctionnent, contiennent le sel LiF. Or, ce sel ne conduit pas toujours au même résultat selon la matrice utilisée. En effet, lorsque ce sel est associé à la matrice CMBT la fluence est égale à 32 % alors qu'avec les matrices 2,4-DHB et HPA, aucun résultat n'est obtenu. Par cette représentation, nous mettons en évidence un comportement particulier des couples avec LiF ce qui nous mène à penser qu'il peut exister une interaction entre le sel LiF et la matrice ou que les résultats de ces expériences doivent être vérifiés.

Pour résumer, on peut classer les couples sel-matrice en trois groupes. Le premier constitué des sels de Rb et K, ne donnent pas de résultats indépendamment du contre ion. Les sels de Na et de Li, constituant le deuxième groupe conduisent à des résultats variables en fonction du contre ion et/ou de la matrice. Le troisième groupe constitué des métaux de transition (Ag et Cu) donnent systématiquement de bons résultats. A ce stade de l'étude, il est difficile de définir un modèle qui rende compte de l'ensemble des expériences MALDI, c'est à dire celles qui fonctionnent et celles qui ne donnent aucun résultat. En effet, nous arrivons à obtenir une discrimination entre sels, mais les conclusions relatives aux descripteurs des matrices restent fragiles. Ce comportement nous amène à nous interroger quant à l'absence de descripteurs considérant à la fois la matrice et le sel.

1.2 Cas d'étude N°2 : étude de solvants

La directive européenne REACH (**R**egistration, **E**valuation, **A**uthorization and restriction of **CH**emicals) (règlement n°1907/2006) entrée en vigueur en 2007 vise à sécuriser la fabrication et l'utilisation de substances chimiques dans l'industrie européenne. Il s'agit de règlementer l'enregistrement, l'évaluation, l'autorisation et les restrictions des substances chimiques. Les principaux objectifs de REACH sont d'assurer la protection de la santé humaine et de l'environnement face aux risques potentiels des produits chimiques, de promouvoir des méthodes d'essais alternatives, de règlementer la libre circulation des substances au sein du marché intérieur et de renforcer la compétitivité et l'innovation de l'industrie. Ainsi, REACH fait porter à l'industrie la responsabilité d'évaluer et de gérer les risques causés par les produits chimiques et de fournir des informations de sécurité adéquates à leurs utilisateurs. Tous les industriels doivent dorénavant enregistrer au niveau européen les substances fabriquées ou importées en quantité supérieure à 1 tonne/an qui après enregistrement conduira à plusieurs décisions :

- la substance est déclarée sans risques et peut être utilisée,
- la substance présente des risques qui peuvent être maitrisés par des précautions d'utilisation et entraine une utilisation sous conditions,
- la substance présente certains risques qui impliquent une utilisation encadrée voire interdite. Dans ce cas, il sera nécessaire de la remplacer par une substance de substitution.

La règlementation REACH demande également de limiter les essais *in vivo* et encourage d'utiliser d'autres méthodes pour évaluer les risques des produits chimiques. Dans ce contexte, les études QSAR sont préconisées pour prévoir l'activité biologique et les propriétés physico-chimiques d'une molécule par une relation mathématique validée.

L'objectif de cette étude est de répondre à la règlementation REACH en trouvant des solvants qui peuvent être éco-compatibles avec le dichlorométhane (DCM), c'est-à-dire ceux qui pourront éviter son utilisation en le remplaçant par un solvant moins dangereux. En effet, l'utilisation du DCM a été règlementée par le Parlement européen qui interdit la mise sur le marché de décapants de peinture contenant du dichlorométhane.

Dans notre étude, nous disposons d'une base de données comptant 236 solvants regroupés dans le tableau 2.4 avec le numéro CAS et la dénomination commune.

Tableau 2.4: Base de données des solvants.

N° du solvant	Numéro CAS	Dénomination commune	N° du solvant	Numéro CAS	Dénomination commune
1	56-23-5	Tétrachlorure de carbone	119	108-86-1	Bromure de phényle
2	56-81-5	Glycérol	120	108-87-2	Méthylcyclohexane
3	57-55-6	Propylène glycol	121	108-88-3	Toluène
4	60-29-7	Diéthyle ether	122	108-89-4	4-picoline
5	62-53-3	Aniline	123	108-90-7	Chlorure de phényle
6	64-17-5	Ethanol	124	108-91-8	Cyclohexylamine
7	64-19-7	Acide acétique	125	108-93-0	Cyclohexanol
8	67-56-1	Méthanol	126	108-94-1	Cyclohexanone

Tableau 2.4: Base de données des solvants.

N° du solvant	Numéro CAS	Dénomination commune	N° du solvant	Numéro CAS	Dénomination commune
9	67-63-0	Propane-2-ol	127	108-95-2	Phénol
10	67-64-1	Acétone	128	108-99-6	3-picoline
11	67-66-3	Chloroforme	129	109-65-9	1-bromobutane
12	67-68-5	Diméthylsulfoxyde	130	109-66-0	n-pentane
13	68-12-2	N,N-diméthylformamide	131	109-69-3	1-chlorobutane
14	71-23-8	1-propanol	132	109-73-9	n-butylamine
15	71-36-3	Alcool butylique	133	109-74-0	Cyanure de propyle
16	71-41-0	1-pentanol	134	109-86-4	2-méthoxyéthanol
17	71-43-2	Benzène	135	109-89-7	Diéthylamine
18	71-55-6	1,1,1-trichloroéthane	136	109-94-4	Formiate d'éthyle
19	74-96-4	Bromoéthane	137	109-99-9	Tétrahydrofuranne
20	75-03-6	Iodure d'éthyle	138	110-12-3	5-méthyl-2-hexanone
21	75-05-8	Acétonitrile	139	110-19-0	Acétate d'isobutyle
22	**75-09-2**	**Dichlorométhane**	140	110-43-0	heptane-2-one
23	75-12-7	Formamide	141	110-54-3	n-hexane
24	75-15-0	Disulfure de carbone	142	110-63-4	1,4-butanediol
25	75-29-6	2-chloropropane	143	110-71-4	1,2-dimethoxyéthane
26	75-34-3	1,1-dichloroéthane	144	110-74-7	Formiate de propyle
27	75-35-4	1,1-dichloroéthylène	145	110-82-7	Cyclohexane
28	75-52-5	Nitrométhane	146	110-83-8	Cyclohexène
29	75-64-9	tert-butylamine	147	110-86-1	Pyridine
30	75-65-0	tert-butanol	148	110-89-4	Pipéridine
31	75-83-2	2,2-diméthylbutane	149	110-91-8	Morpholine
32	75-85-4	2-méthylbutane-2-ol	150	111-13-7	Octane-2-one
33	75-89-8	2,2,2-trifluoroéthanol	151	111-27-3	1-hexanol
34	75-97-8	3,3-diméthyl-2-butanone	152	111-42-2	Diéthanolamine
35	76-05-1	Acide trifluoroacétique	153	111-43-3	Ether dipropylique
36	78-59-1	Isophorone	154	111-46-6	Diéthylène glycol
37	78-81-9	Isobutylamine	155	111-55-7	Ethylene glycol diacétate
38	78-83-1	Isobutanol	156	111-65-9	n-octane
39	78-86-4	Chlorure de sec-butyle	157	111-70-6	heptan-1-ol
40	78-87-5	1,2-dichloropropane	158	111-76-2	2-butoxyethanol
41	78-92-2	2-butanol	159	111-84-2	n-nonane
42	78-93-3	Butanone	160	111-87-5	1-octanol
43	79-00-5	1,1,2-trichloroéthane	161	111-96-6	Oxyde de bis(2-méthoxyéthyle)
44	79-01-6	1,1,2-trichloroéthylène	162	112-27-6	Triéthylène glycol

Tableau 2.4: Base de données des solvants.

N° du solvant	Numéro CAS	Dénomination commune	N° du solvant	Numéro CAS	Dénomination commune
45	79-16-3	N-méthylacétamide	163	112-30-1	1-décanol
46	79-20-9	Acétate de méthyle	164	112-36-7	Diéthylèneglycol diéthyléther
47	79-24-3	Nitroéthane	165	112-53-8	Dodécanol
48	79-29-8	2,3-diméthylbutane	166	112-60-7	Tétraéthylène glycol
49	79-34-5	1,1,2,2-tétrachloroéthane	167	119-36-8	Méthyl salicylate
50	79-46-9	2-nitropropane	168	119-64-2	Tétraline
51	90-12-0	1-méthylnaphtalène	169	120-82-1	1,2,4-trichlorobenzène
52	91-17-8	Décahydronaphtalène	170	120-92-3	Cyclopentanone
53	91-22-5	Quinoléine	171	121-44-8	Triéthylamine
54	93-58-3	Benzoate de méthyle	172	121-69-7	N,N-diméthylaniline
55	93-89-0	Benzoate d'éthyl	173	123-19-3	Heptane-4-one
56	95-47-6	o-xylène	174	123-39-7	N-méthylformamide
57	95-48-7	o-cresol	175	123-42-2	4-hydroxy-4-méthylpentane-2-one
58	95-50-1	o-dichlorobenzène,	176	123-51-3	Alcool isoamylique
59	96-14-0	3-méthylpentane	177	123-54-6	2,4-pentanedione
60	96-22-0	3-pentanone	178	123-72-8	Butyraldéhyde
61	96-37-7	Méthylcyclopentane	179	123-75-1	Pyrrolidine
62	96-48-0	Butyrolactone	180	123-86-4	Acétate de n-butyle
63	96-49-1	Carbonate d'éthylène	181	123-91-1	1,4-dioxane
64	97-95-0	2-ethyl-1-butanol	182	123-92-2	Acétate d'isoamyle
65	98-00-0	Alcool furfurylique	183	123-96-6	2-octanol
66	98-01-1	2-furaldéhyde	184	124-18-5	Décane
67	98-82-8	Isopropylbenzène	185	126-33-0	Sulfolane
68	98-86-2	Acétophénone	186	127-18-4	1,1,2,2-tétrachloroéthylène
69	98-87-3	Chlorure de benzylidène	187	127-19-5	N,N-diméthylacétamide
70	98-95-3	Nitrobenzène	188	140-29-4	Cyanure de benzyle
71	100-37-8	2-diéthylaminoethanol	189	140-88-5	Acrylate d'éthyle
72	100-41-4	Ethylbenzène	190	141-43-5	Ethanolamine
73	100-42-5	Styrène	191	141-78-6	Acétate d'éthyle
74	100-47-0	Cyanure de phényle	192	141-79-7	Oxyde de mésityle
75	100-51-6	Alcool benzylique	193	142-29-0	Cyclopentène
76	100-52-7	Benzaldehyde	194	142-68-7	Tétrahydropyrane
77	100-66-3	Oxyde de méthyle et de phényle (anisole)	195	142-82-5	n-heptane
78	101-84-8	oxyde de diphényle	196	142-84-7	Dipropylamine

Tableau 2.4: Base de données des solvants.

N° du solvant	Numéro CAS	Dénomination commune	N° du solvant	Numéro CAS	Dénomination commune
79	102-71-6	Triéthanolamine	197	142-92-7	Acétate d'hexyle
80	102-82-9	Tributylamine	198	142-96-1	Oxyde de dibutyle
81	103-50-4	Oxyde de dibenzyle	199	143-08-8	Nonanol
82	103-73-1	Ethoxybenzène	200	143-24-8	Tétraéthylène glycol diméthyléther
83	104-51-8	Butylbenzène	201	156-59-2	cis-1,2-dichloroéthylène
84	105-05-5	1,4-diéthylbenzène	202	287-92-3	Cyclopentane
85	105-37-3	Propionate d'éthyl	203	352-93-2	Sulfure de diéthyle
86	105-39-5	Chloroacétate d'éthyle	204	462-06-6	Fluorobenzène
87	105-57-7	1,1-diéthoxyéthane	205	504-63-2	Triméthylèneglycol
88	105-58-8	Carbonate de diéthyle	206	512-56-1	Phosphate de méthyle
89	106-35-4	Heptane-3-one	207	540-54-5	Chlorure de propyle
90	106-42-3	p-xylène	208	540-59-0	1,2-dichloroéthylène
91	106-93-4	1,2-dibromoéthane	209	540-84-1	2,2,4-triméthylpentane
92	107-06-2	1,2-dichloroéthane	210	541-73-1	m-dichlorobenzène
93	107-07-3	2-chloroéthanol	211	545-06-2	Trichloroacétonitrile
94	107-10-8	Propylamine	212	563-80-4	3-méthyl-2-butanone
95	107-12-0	Cyanure d'éthyle	213	565-80-0	2,4-diméthyl-3-pentanone
96	107-15-3	1,2-diaminoéthane	214	576-26-1	2,6-diméthylphénol
97	107-18-6	Alcool allylique	215	584-02-1	3-pentanol
98	107-21-1	Ethane-1,2-diol	216	591-50-4	Iodure de phényle
99	107-31-3	Formiate de méthyle	217	591-78-6	Hexane-2-one
100	107-41-5	Hexylene glycol	218	592-41-6	1-hexène
101	107-83-5	2-méthylpentane	219	592-76-7	Hept-1-ène
102	107-87-9	pentane-2-one	220	616-38-6	Carbonate de diméthyle
103	107-88-0	butane-1,3-diol	221	616-45-5	2-pyrrolidinone
104	107-92-6	Acide butyrique	222	628-63-7	Acétate de pentyle
105	108-03-2	1-bitropropane	223	629-14-1	1,2-diéthoxyéthane
106	108-10-1	4-méthylpentane-2-one	224	632-22-4	1,1,3,3-tétraméthylurée
107	108-18-9	Diisopropylamine	225	680-31-9	Hexaméthylphosphoramide
108	108-20-3	Ddiisopropyléther	226	685-91-6	N,N-diéthylacétamide
109	108-21-4	Acétate d'isopropyle	227	872-50-4	N-méthylpyrrolidone
110	108-24-7	Anhydride acetique	228	1119-40-0	Glutarate de diméthyle
111	108-32-7	Carbonate de propylène	229	1634-04-4	Oxyde de tert-butyle et de méthyle
112	108-38-3	m-xylène	230	2807-30-9	2-(propyloxy)éthanol
113	108-39-4	m-cresol	231	5989-27-5	d-limonène

Tableau 2.4: Base de données des solvants.

N° du solvant	Numéro CAS	Dénomination commune	N° du solvant	Numéro CAS	Dénomination commune
114	108-47-4	2,4-diméthylpyridine	232	6032-29-7	Alcool sec-amylique
115	108-48-5	2,6-diméthylpyridine	233	7226-23-5	1,3-diméthyl-2-oxohexahydropyrimidine
116	108-67-8	mésitylène	234	13952-84-6	sec butylamine-R
117	108-75-8	2,4,6-triméthylpyridine	235	29911-28-2	Dowanol DPnB
118	108-83-8	2,6-diméthylheptane-4-one	236	69411-44-9	Méthylpropylbenzène-m

Ces 236 solvants sont décrits par 11 descripteurs issus de calculs théoriques de la chimie quantique :

- l'énergie de la plus haute orbitale moléculaire occupée (HOMO)
- l'énergie de la plus basse orbitale moléculaire vacante (LUMO)
- le moment dipolaire
- la polarisabilité (α)
- la charge atomique maximale de Mülliken (q_{max})
- la charge atomique minimale de Mülliken (q_{min})
- le potentiel électrostatique le plus haut (V_{max})
- le potentiel électrostatique le plus bas (V_{min})
- la surface
- le potentiel électrostatique inférieur à $-0.1eV$
- le potentiel électrostatique compris entre -0.1 et $+0.1eV$

Pour trouver un solvant qui pourrait substituer le DCM, dans un premier temps nous devons identifier, parmi les solvants de la base de données, ceux qui présentent des caractéristiques similaires au DCM du point de vue des descripteurs. Pour cela, nous proposons l'ACC pour visualiser les solvants de la base de données décrits par les 11 descripteurs. Ainsi, si nous recherchons sur la projection ACC, les solvants voisins du DCM, nous pourrons les proposer comme substituants. La figure 2.5 présente la projection ACC.

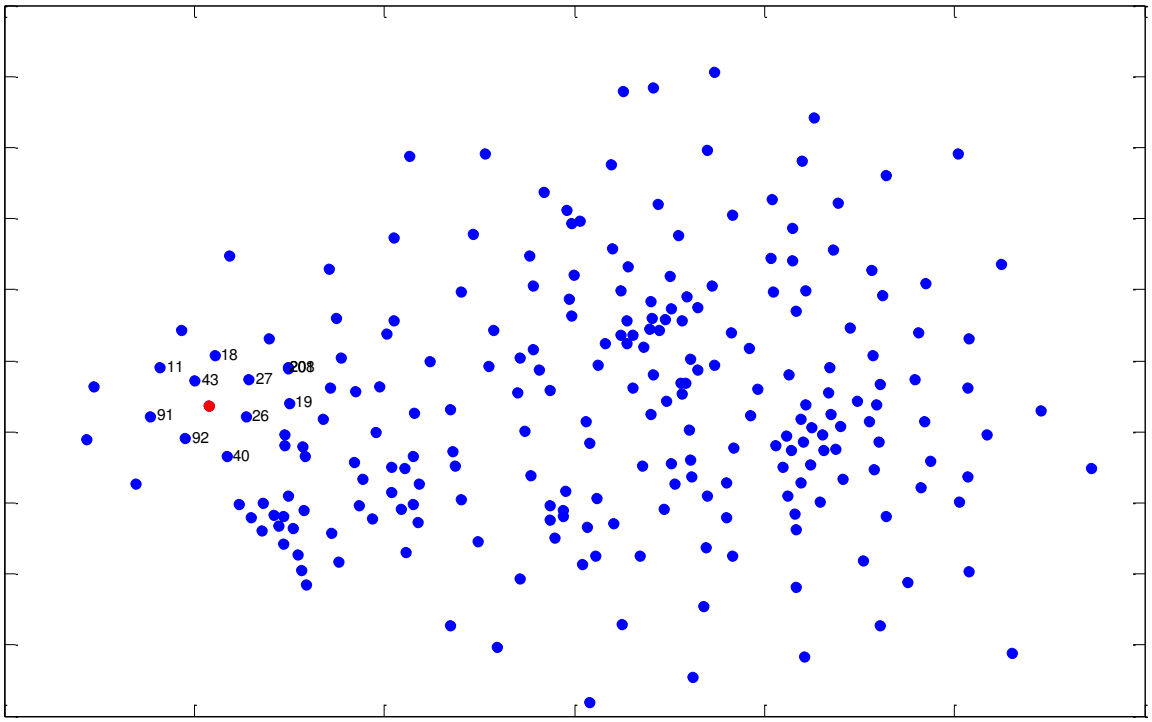


FIGURE 2.5 – Projection ACC de la base de données constituée par les 236 solvants décrits par les 11 descripteurs. Le point rouge représente le dichlorométhane et les points numérotés correspondent aux plus proches voisins.

Pour une identification plus facile, nous effectuons un zoom de la projection ACC représenté par la figure 2.6, autour du DCM.

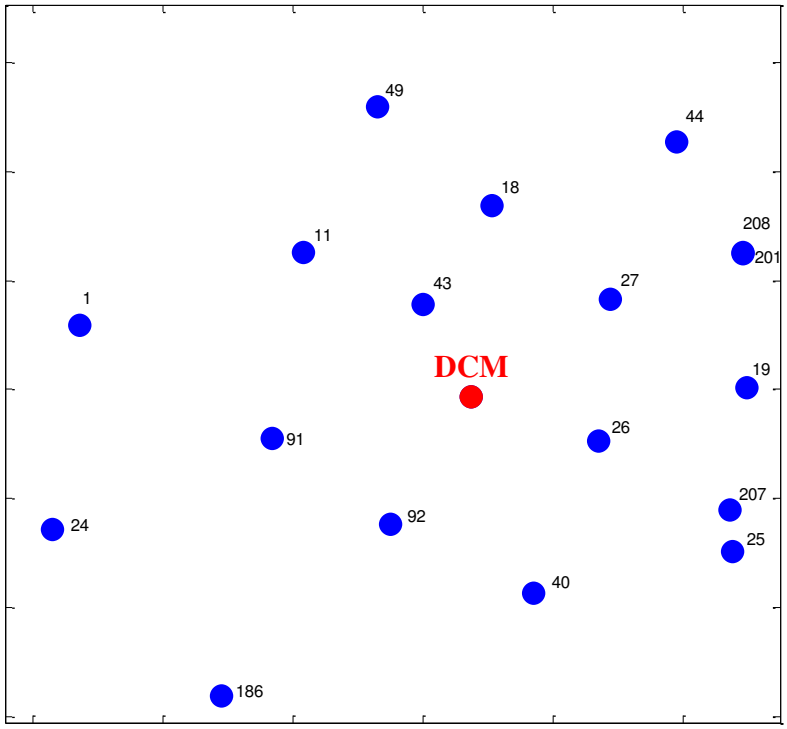


FIGURE 2.6 – Solvants situés à proximité du dichlorométhane sur la projection ACC.

Sur la figure 2.6, nous pouvons clairement identifier les solvants situés à proximité du DCM (représenté en rouge) et nous trouvons :

- le point n°43 : 1,1,2-trichloroéthane,
- le point n°26 : 1,1-dichloroéthane,
- le point n°92 : 1,2-dichloroéthane,
- le point n°27 : 1,1-dichloroéthylène,
- le point n°91 : 1,2-dibromoéthane,
- le point n°18 : 1,1,1-trichloroéthane,
- le point n°11 : chloroforme,
- le point n°40 : 1,2-dichloropropane,

et dans une moindre mesure les points n°19, 201 et 208 qui correspondent respectivement au bromoéthane, cis-1,2-dichloroéthylène, 1,2-dichloroéthylène. Cette première analyse nous permet de visualiser les solvants situés dans la sphère de proximité du DCM.

Nous proposons de compléter ces observations graphiques par une deuxième approche en travaillant non plus dans un espace de projection pour lequel on ne considère pas la totalité de l'information mais dans l'espace à 11 dimensions des descripteurs. Cette approche consiste à utiliser l'algorithme WSP pour identifier les solvants voisins du DCM. Pour cela, les solvants sont représentés par des points dans l'espace des variables et nous admettrons que des points proches correspondent à des solvants au comportement similaire. D'un point de vue pratique, nous allons imposer le DCM comme point de départ de l'algorithme WSP, calculer les distances euclidiennes avec tous les autres solvants, puis progressivement nous allons faire varier la valeur limite (d_{min}) en deçà de laquelle les points sont considérés comme proches en information. Ici, nous nous intéressons uniquement à la première étape de l'algorithme à savoir identifier les solvants les plus proches du DCM, en débutant par une très faible valeur d_{min}.

Lorsque $d_{min} = 0.2$, la première étape de l'algorithme WSP ne retient aucun point ce qui signifie que l'hypersphère de rayon 0.2 centrée sur le DCM ne contient aucun autre solvant. Si nous fixons une valeur $d_{min} = 0.25$, seul le solvant n°43 est retenu, c'est donc ce solvant qui présente le plus de similarités avec le DCM. Ainsi, l'augmentation progressive de la valeur d_{min} nous permet d'identifier les solvants de manière itérative. Si nous augmentons la valeur d_{min} à 0.30, trois solvants sont identifiés : chloroforme (n°11), 1,1-dichloroéthane (n°26) et 1,1,2-trichloroéthane (n°43). Si $d_{min} =$ 0.35, 8 solvants sont retenus : chloroforme (n°11), 1,1,1-trichloroéthane (n°18), 1,1-dichloroéthane (n°26), 1,1-dichloroéthylène (n°27), 1,1,2-trichloroéthane (n°43), 1,2-dichloroéthane (n°92), cis-1,2-dichloroéthylène (n°201) et 1,2-dichloroéthylène (n°208).

La figure 2.7 regroupe ces informations issues de l'algorithme WSP pour différentes valeurs d_{min}, en représentant les sphères de proximité des solvants par rapport au DCM. Le rayon de chacune des sphères est égal aux valeurs d_{min} que nous avons testées et qui ont conduit à l'identification progressive des solvants en fonction de leur proximité avec le DCM. Nous précisons que cette figure doit être interprétée avec précaution car elle permet d'identifier uniquement les solvants voisins du DCM et ne renseigne pas sur la proximité des autres solvants entre eux.

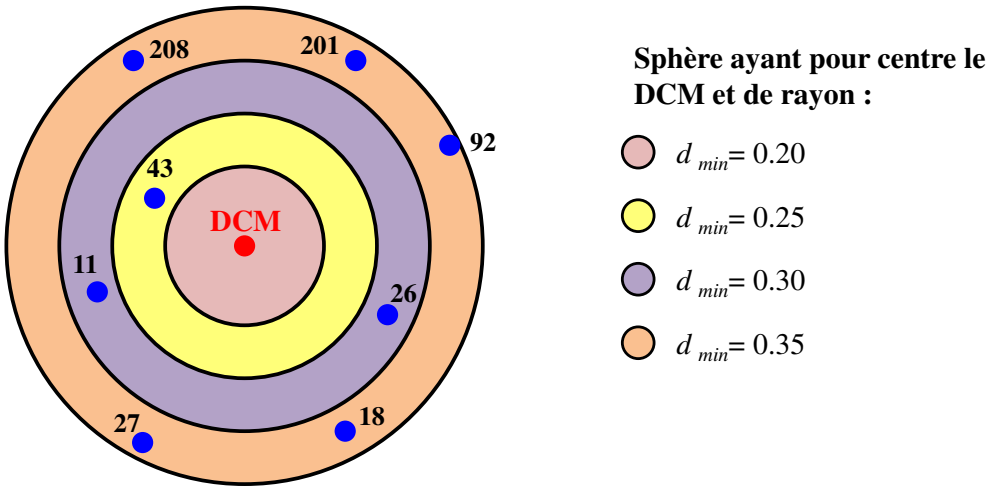


FIGURE 2.7 – Schéma représentant les solvants voisins du DCM.

A partir de la base de données constituée par les 236 solvants nous avons appliqué consécutivement l'ACC et l'algorithme WSP. Par ces deux approches complémentaires, nous identifions les mêmes solvants présentant de fortes similarités avec le DCM pour les descripteurs utilisés. L'ACC permet de visualiser ces solvants dans un espace en 2D et l'algorithme WSP est utilisé non pas pour sélectionner des points mais pour identifier les points les plus proches du DCM.

1.3 Cas d'étude N°3 : étude de l'excès énantiomérique

Dans cette étude, nous nous intéressons à une réaction d'hydrolyse enzymatique de diesters *meso* par l'estérase de foie de porc (PLE). Nous souhaitons modéliser l'excès énantiomérique en fonction des descripteurs caractérisant les structures des substrats de l'hydrolyse enzymatique. Trois familles de composés ont été étudiées, dont une série de malonates pour lesquels la réaction est présentée figure 2.8 :

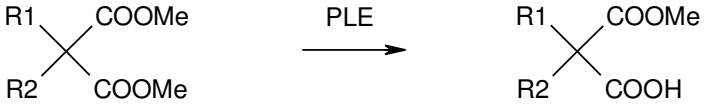
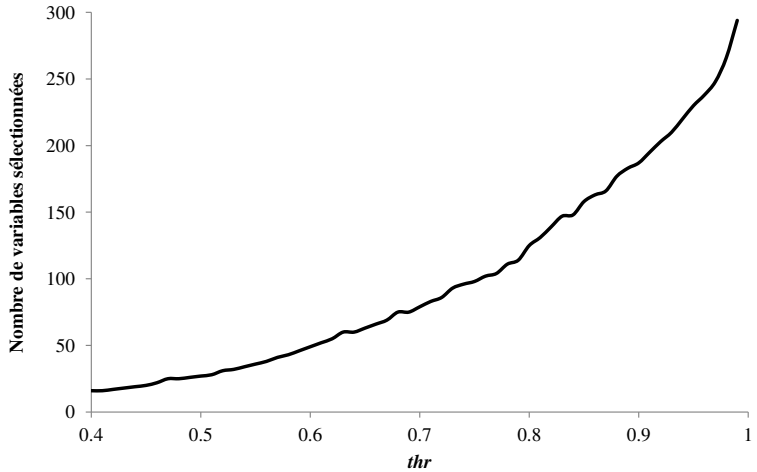


FIGURE 2.8 – Réaction de l'hydrolyse enzymatique de diesters *meso* par la PLE.

Pour étudier cette hydrolyse enzymatique, 26 substrats ont été caractérisés par les descripteurs moléculaires correspondant à la géométrie de plus basse énergie, calculés à l'aide du logiciel Dragon [73, 74, 75].

L'objectif de cette étude QSAR est donc de construire un modèle prévisionnel pour évaluer l'excès énantiomérique (ee) du mono-acide formé lors de l'hydrolyse enzymatique. Pour cela, nous disposons d'un jeu de données constitué par les 26 substrats (lignes) pour lesquels 388 descripteurs moléculaires (colonnes) ont été calculés. Ainsi, notre base de données compte beaucoup moins de lignes que de colonnes ce qui impose de sélectionner un sous-ensemble de descripteurs parmi les 388 pour construire le modèle QSAR. Pour ce faire, nous proposons d'utiliser l'algorithme V-WSP qui permet de sélectionner les variables les plus explicatives en éliminant les variables corrélées ou redondantes.

Dans la première partie de ce manuscrit, nous avons montré que l'utilisation de l'algorithme V-WSP pour le choix des variables demande de définir une valeur de la limite de corrélation *thr* et la variable de référence. Dans un premier temps, nous proposons de faire varier la valeur *thr* entre 0.4 et 1 et de suivre l'évolution du nombre de variables sélectionnées et du critère *procrustes*. Pour une valeur *thr* donnée, nous obtiendrons autant de solutions que de descripteurs et il est intéressant de représenter le nombre moyen de variables sélectionnées à partir des 388 solutions en fonction de la valeur *thr* (figure 2.9).



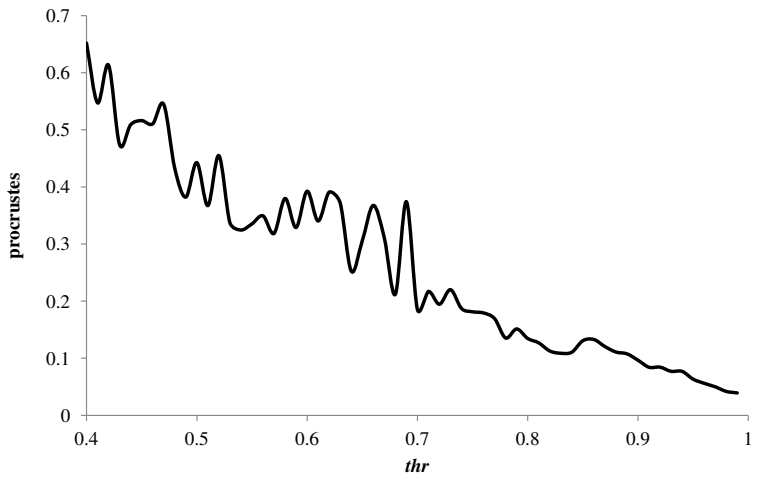


FIGURE 2.9 – Évolution du nombre de variables sélectionnées et du critère *procrustes* en fonction de la valeur de la limite de corrélation *thr*.

Sur la figure 2.9, nous observons que plus la valeur thr est élevée plus le nombre de variables sélectionnées est important et plus la valeur du critère *procrustes* est faible. Nous rappelons qu'une faible valeur *procrustes* implique que la structure réduite est similaire à la structure initiale. Le choix de la valeur thr conditionne le nombre de variables choisies. En effet, lorsque nous fixons $thr = 0.4$, 16 variables sont sélectionnées et le critère *procrustes* est élevé ($= 0.65$) alors que si nous augmentons la limite thr à 0.9, l'algorithme V-WSP retient 187 descripteurs et conduit à une faible valeur $procrustes = 0.1$. Or dans ce cas, pour construire notre modèle, ayant seulement 26 substrats à notre disposition, nous ne pourrons pas sélectionner plus de 25 descripteurs (avec un modèle de degré 1). Ceci implique de fixer une valeur thr faible telle que : $0.4 \leq thr \leq 0.5$.

Pour sélectionner les variables les plus pertinentes en respectant la contrainte imposée par le nombre de substrats, nous envisageons différentes stratégies, sachant que nous disposons des résultats expérimentaux (excès énantiomérique) pour les 26 substrats :

- fixer une faible valeur de thr en considérant le critère *procrustes* et ne retenir que les variables communes à toutes les solutions,
- sélectionner le sous-ensemble de variables par stepwise.

Comme la solution obtenue par l'algorithme V-WSP dépend de la valeur thr mais aussi de la variable de référence, nous obtiendrons 388 solutions pour une valeur thr donnée. Nous choisissons de ne retenir que celle qui présente la plus faible valeur du critère *procrustes*.

1.3.1 Sélection des variables pour une faible valeur de *thr*

Comme présenté ci-dessus, la première approche consiste à fixer une faible valeur pour la limite de corrélation thr.

Nous choisissons de fixer deux valeurs thr dans l'intervalle imposé : pour $thr = 0.40$, nous obtenons une solution à 14 variables avec la plus faible valeur *procrustes* ($= 0.298$), et pour $thr = 0.43$, la meilleure solution compte 18 variables pour un critère $procrustes = 0.275$.

Pour quantifier la relation entre l'excès énantiomérique et les descripteurs sélectionnés par V-WSP, nous proposons de postuler un modèle du premier degré de la forme équation (1.1) :

$$Y = b_0 + \sum_{i=1}^{D'} b_i X_i \tag{1.1}$$

avec X_i les D' variables sélectionnées par V-WSP et $(b_j)_{j\in\{0,1,\ldots,D'\}}$ les coefficients du modèle.

Pour valider les modèles, nous proposons de calculer les critères *a posteriori* suivants :

– le coefficient de détermination R^2 qui mesure la qualité de l'ajustement :

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

avec y_i la réponse expérimentale (ici l'excès énantiomérique), $\hat{y}_i$ la réponse calculée par le modèle, $\bar{y}$ la valeur moyenne de la réponse et N le nombre de substrats.

– le coefficient de détermination ajusté $R^2 ajusté$, qui contrairement au R^2 permet de s'affranchir du nombre de variables :

$$R^2 ajusté = R^2 - \frac{D'(1-R^2)}{N - D' - 1}$$

avec D' le nombre de variables sélectionnées par l'algorithme V-WSP

– la racine carrée de l'erreur quadratique moyenne RMSE :

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

– l'erreur absolue maximale :

$$MAX = max\,\|y_i - \hat{y}_i\|$$

Le tableau 2.5 regroupe les valeurs des critères *a posteriori* du modèle calculé pour les deux solutions à 14 et 18 variables obtenues respectivement pour $thr = 0.40$ et 0.43 :

Tableau 2.5 – Estimation de la qualité du modèle pour l'excès énantiomérique.

thr	*procrustes*	Nombre de variables sélectionnées par V-WSP	R^2	$R^2 ajusté$	*RMSE*	*MAX*	Nombre de degrés de liberté
0.40	0.298	14	0.826	0.605	13.82	34.62	11
0.43	0.275	18	0.824	0.372	13.89	36.17	7

Les valeurs regroupées dans le tableau 2.5 permettent d'observer que les deux solutions, obtenues pour des valeurs *thr* proches avec des valeurs *procrustes* similaires (0.298 et 0.275), présentent des valeurs de critère $R^2 ajusté$ différentes (0.605 et 0.372) et trop faibles pour les modèles calculés par régression des moindres carrés. Cette première approche ne permet pas d'obtenir des modèles de bonne qualité probablement du fait de la faible valeur *thr* qui permet de ne retenir que 4% des variables initiales ce qui n'est pas suffisant.

1.3.2 Sélection des variables communes à toutes les solutions de V-WSP

Nous rappelons que pour une valeur *thr* l'algorithme V-WSP propose autant de solutions que de variables. Dans cette deuxième approche, nous proposons d'identifier les variables communes à toutes les solutions, pour chaque valeur *thr*. Nous représentons figure 2.10 l'évolution du nombre de variables communes à toutes les solutions et le critère *procrustes* en fonction de la valeur *thr* (courbes bleues). Afin de comparer cette approche à celle précédemment évoquée, nous avons ajouté sur les graphes de la figure 2.10, les résultats de la première approche que nous représentons par les courbes noires.

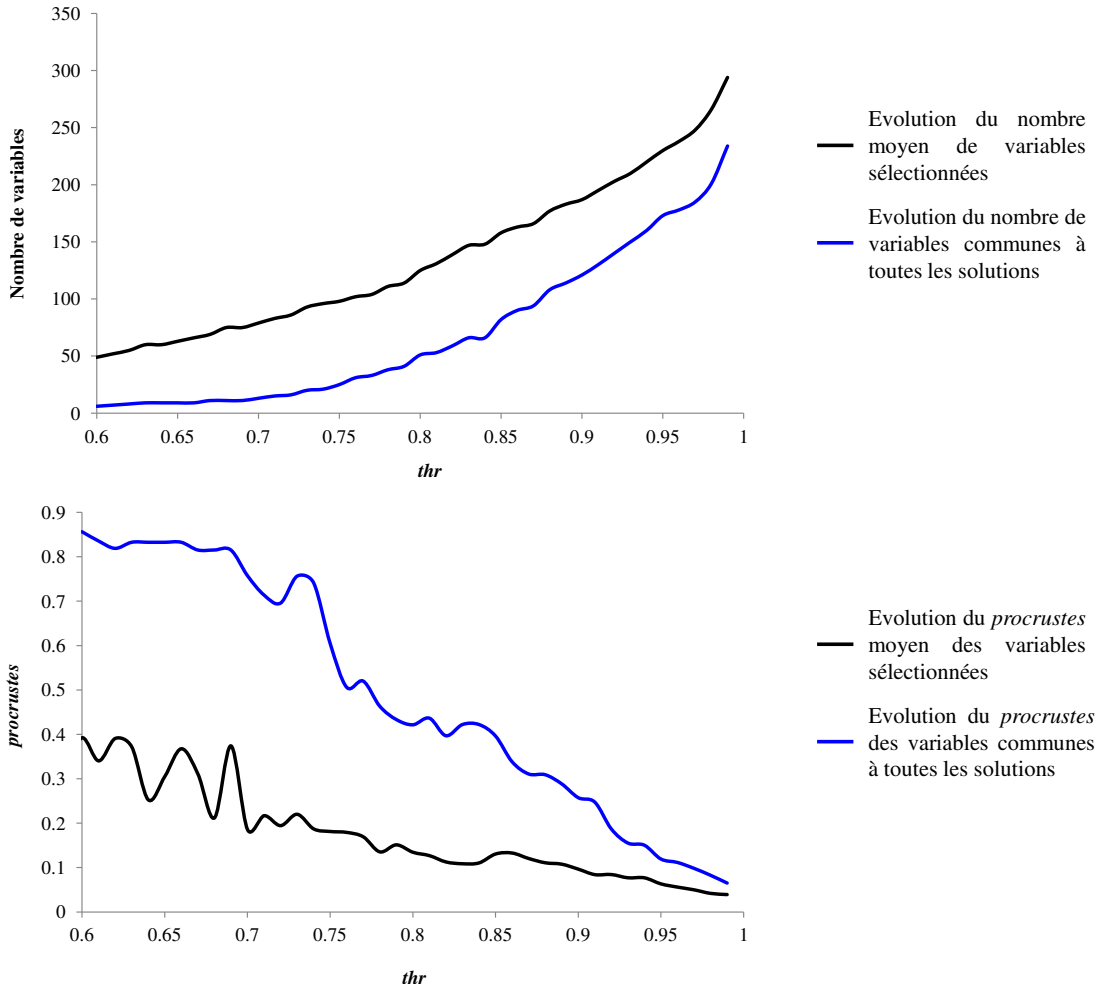


FIGURE 2.10 – Évolution du nombre de variables communes à toutes les solutions et du critère *procrustes* des structures réduites en fonction de la valeur *thr*.

Nous observons que dans cette deuxième stratégie, le nombre de variables communes pour une valeur *thr* fixée (courbe bleue) suit la même tendance que le nombre moyen de variables sélectionnées selon la première approche (courbe noire) mais s'accompagne d'une forte augmentation du critère *procrustes* pour des faibles valeurs *thr*. Le tableau 2.6 présente quelques valeurs du critère *procrustes* calculées à partir des variables communes à toutes les solutions pour des valeurs $thr = 0.7, 0.8, 0.82, 0.9$.

Tableau 2.6 – Évolution du critère *procrustes* en fonction des variables communes à toutes les solutions pour différentes valeurs de *thr*.

thr	Nombre de variables communes à toutes les solutions	*procrustes*
0.7	13	0.758
0.8	51	0.422
0.82	59	0.397
0.9	121	0.257

Lorsque la valeur *thr* augmente de 0.7 à 0.9, le nombre de variables communes augmente de 13 à 121 ce qui induit une diminution du *procrustes* de 0.758 à 0.257. Si nous nous intéressons davantage aux variables communes nous constatons que l'augmentation du *thr* ne fait qu'ajouter de nouvelles variables à la solution précédente. En d'autres termes, si nous reprenons les solutions reportées dans le tableau 2.6, les 13 variables communes obtenues pour $thr = 0.7$ sont retrouvées dans la solution résultant de $thr = 0.8$ qui elle même est contenue dans les 59 variables communes pour $thr = 0.82$.

L'idée d'extraire un nouveau sous-ensemble à partir des variables sélectionnées par l'algorithme V-WSP nous a semblé intéressante mais avec cette deuxième approche, les variables communes ne permettent pas d'obtenir de bons critères *procrustes* lorsque le nombre de descripteurs retenus est faible, ce qui nous empêche d'envisager l'étape de modélisation par régression des moindres carrés.

1.3.3 Sélection des variables les plus représentatives par régression stepwise

Dans cette troisième approche nous proposons, pour une valeur *thr* fixée, de ne retenir que la solution présentant la plus faible valeur *procrustes*, puis de sélectionner les variables les plus significatives par régression stepwise. Nous rappelons que la régression stepwise qui consiste à introduire progressivement les variables qui entrainent un accroissement significatif du R^2 permet de sélectionner D'' variables explicatives parmi les D' variables sélectionnées par V-WSP (avec $D'' < D'$).

Pour trois valeurs *thr*, 0.7, 0.8, 0.9, nous avons représenté figure 2.11 l'évolution du coefficient de détermination R^2 en fonction du nombre de variables ajoutées par stepwise dans le modèle. Les sous-ensembles initiaux comptent respectivement : 79, 127 et 192 variables avec de faibles valeurs de *procrustes* égales à 0.158, 0.109, 0.087.

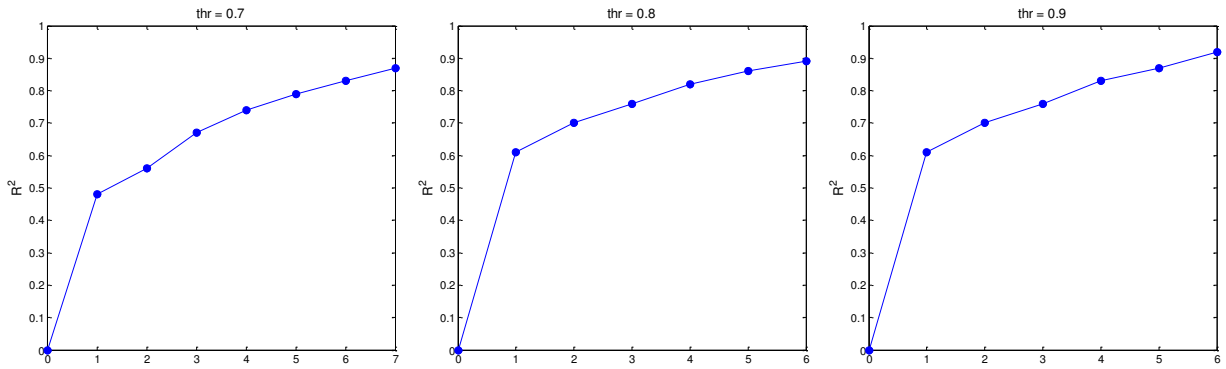


FIGURE 2.11 – Évolution du coefficient de détermination R^2 en fonction du nombre de variables ajoutées par stepwise en considérant trois valeurs de la limite de corrélation *thr* : 0.70 ; 0.80 et 0.90.

Nous observons que lorsque la limite de corrélation est fixée à $thr = 0.7$, 7 variables seront retenues alors que nous en retiendrons 6 pour $thr = 0.8$ et $thr = 0.90$. A partir de ces sous-ensembles de variables, nous postulons deux types de modèle. Le premier est un modèle additif et le deuxième un modèle PLS (Partial Least Squares) [76, 77] avec trois composantes. Le tableau 2.7 regroupe les valeurs des critères *a posteriori* obtenus pour les deux modèles envisagés.

Tableau 2.7 – Critères *a posteriori* des modèles obtenus en considérant les variables sélectionnées par régression stepwise.

	Modèle additif			**Modèle PLS**		
thr	0.7	0.8	0.9	0.7	0.8	0.9
Nombre de variables issues de l'algorithme V-WSP	79	127	192	79	127	192
Nombre de variables retenues par stepwise	7	6	6	7	6	6
R^2	0.844	0.889	0.923	0.811	0.888	0.902
$R^2 ajusté$	0.783	0.853	0.899	0.738	0.853	0.871
RMSE	13.1	11.06	9.2	14.4	11.08	10.4
MAX	44.79	22.05	19.17	47	21.87	24.5
Nombre de degrés de liberté	18	19	19	18	19	19

Les résultats du tableau 2.7 montrent que les critères *a posteriori* des deux modèles conduisent à des valeurs proches voire une légère préférence pour le modèle linéaire de degré 1. Par ailleurs, quel que soit le modèle postulé, les meilleurs critères sont obtenus lorsque la régression stepwise est effectuée sur les 192 variables sélectionnées par l'algorithme V-WSP en fixant $thr = 0.90$.

Par cette approche nous sommes alors capables de proposer un modèle contenant 6 descripteurs avec une valeur $R^2 ajusté$ égale à 0.9.

Dans le cadre de cette étude QSAR, nous avons utilisé l'algorithme V-WSP pour sélectionner des descripteurs parmi un grand nombre. Trois approches ont été envisagées. La première, qui consiste à fixer une faible limite de corrélation *thr* pour ne retenir que quelques descripteurs, conduit à des modèles de qualité médiocre. La deuxième approche, qui considère les variables communes à toutes les solutions de V-WSP, ne semble pas pertinente au vu de ses résultats. L'utilisation de la régression stepwise pour extraire les variables les plus explicatives à partir de variables non corrélées sélectionnées par l'algorithme V-WSP permet d'obtenir des modèles de bonne qualité. En effet, cette troisième approche conduit à une sélection optimale des variables, en réalisant une première sélection de variables les plus orthogonales possible, pour ensuite ne retenir que les plus explicatives de la variation de la réponse.

1.4 Conclusion

Dans ce chapitre, nous avons traité des données issues des études de relations structure-activité sur lesquelles nous avons appliqué les méthodes présentées dans la première partie de ce manuscrit à savoir l'ACC pour la visualisation des données, l'algorithme WSP pour la sélection de points et l'algorithme V-WSP pour la sélection de variables. En QSAR, la principale difficulté réside dans le grand nombre de descripteurs ce qui induit des bases de données en grande dimension. Par ces méthodes, nous avons proposé une alternative pour le traitement de ce type de données à partir de trois applications.

Dans un premier temps, nous avons proposé d'utiliser l'ACC pour visualiser des couples sel-matrice dans un espace en deux dimensions afin d'identifier ceux qui sont proches dans l'espace de projection et qui présentent, donc, des propriétés similaires alors que ceux qui sont diamétralement opposés peuvent avoir des comportements très différents. Ainsi, nous avons pu mettre en exergue des comportements particuliers dus à l'utilisation de certains sels mais aussi des couples qui présentent des propriétés similaires quels que soient la matrice et le sel envisagés.

Dans la deuxième application, l'objectif était d'identifier un solvant avec des propriétés analogues au dichlorométhane et qui pourrait le substituer. Pour cela, nous avons proposé deux approches : la première est l'ACC qui permet d'identifier les solvants situés à proximité du dichlorométhane et qui présentent donc des propriétés similaires. La deuxième approche utilise l'algorithme WSP pour identifier les solvants qui se trouvent dans la même sphère de solubilité du dichlorométhane. Par ces deux approches, nous avons identifié les mêmes solvants voisins du DCM.

Dans la troisième et dernière application, la base de données compte peu de substrats par rapport au nombre élevé de descripteurs. Ainsi, la difficulté réside dans le grand nombre de variables ce qui nous a conduit à utiliser l'algorithme V-WSP pour sélectionner les variables les plus pertinentes et par la suite obtenir un modèle de bonne qualité. Or, dans cet exemple le faible nombre de substrats nous permet de conserver que très peu de variables ce qui nous a mené à compléter le V-WSP par une régression stepwise afin d'extraire les variables les plus explicatives d'une solution V-WSP.

A partir de ces exemples, nous avons montré que nous disposons de méthodes qui permettent de s'affranchir des conséquences de la grande dimension des données QSAR. En effet, la visualisation des données est rendue possible par l'ACC et la surabondance d'information contenue par le grand nombre de descripteurs (colonnes) peut être réduite en utilisant l'algorithme V-WSP. Ces premiers résultats semblent prometteurs et les méthodes utilisées sont faciles et rapides à mettre en œuvre.

Chapitre 2

Analyse des données spectroscopiques

La spectroscopie est une méthode analytique qui permet d'identifier la composition et la structure de la matière à partir des spectres obtenus par l'interaction de cette matière avec les différents rayonnements électromagnétiques qui sont émis, absorbés ou diffusés. Cette technique est aujourd'hui couramment utilisée dans de nombreux domaines tels que la chimie, la physique, la biologie, l'agroalimentaire, ... afin de sonder une matière et d'en déduire les informations structurales. En fonction du mode d'interaction entre la matière et le rayonnement nous distinguons différents types de spectroscopies :

- la spectroscopie d'absorption qui repose sur l'excitation après absorption d'un photon,
- la spectroscopie d'émission qui émet un photon lors du retour à l'état fondamental (relaxation),
- la spectroscopie de diffusion (Raman) pour laquelle les interactions entre la matière et les radiations électromagnétiques conduisent à des phénomènes de diffusion élastique ou inélastique. Cette diffusion peut avoir lieu à la rencontre d'une interface entre deux milieux ou à la traversée d'un milieu.

Parmi ces approches, la spectroscopie infrarouge suscite un intérêt croissant dans de nombreux domaines d'application tels que l'industrie chimique, l'industrie pharmaceutique, l'agroalimentaire, l'industrie textile, l'environnement, ... La partie infrarouge (IR) du spectre électromagnétique se divise en trois parties : le proche IR (NIR, *Near IR*) pour des longueurs d'onde (λ) comprises entre 0.7μm à 3 μm, le moyen IR (MIR, *Mid IR*) pour 3 μm $< \lambda <$ 25 μm et le lointain IR (FIR, *Far IR*) pour $\lambda >$ 25μm. Le moyen et le proche IR sont couramment utilisés pour l'analyse fonctionnelle ou structurelle de la matière alors que le lointain IR est difficile à utiliser à cause de la faiblesse des sources. La spectrométrie infrarouge exploite le fait que l'énergie du rayonnement IR est suffisante pour modifier les vibrations des molécules et mesure par la suite l'absorption ce rayonnement IR par la matière, ce qui en fait une technique fiable et robuste qui s'utilise principalement pour identifier la nature des liaisons chimiques composant la molécule. Toutefois, devant la forte demande de réduction du temps d'analyse et de simplification des procédés, les appareils de mesure deviennent de plus en plus compétitifs et gagnent en précision en multipliant le nombre de mesures pour un spectre donné. Ainsi, les bases de données spectroscopiques sont de plus en plus conséquentes et nécessitent des outils pour faciliter l'interprétation de ces grandes bases de données.

Pour ce faire, nous proposons d'utiliser les méthodes présentées dans la première partie de ce manuscrit afin de s'affranchir une nouvelle fois des conséquences de la grande dimension. Dans ce chapitre, nous proposons deux études utilisant des données spectroscopiques IR. La première est issue de l'industrie agroalimentaire et cherche à regrouper des échantillons avec le même profil infrarouge, ce qui fait intervenir une notion de proximité et nous conduit à utiliser l'ACC. La deuxième étude, cherche à établir un modèle prévisionnel et nécessite de sélectionner dans la grande base de données, des spectres pour la calibration du modèle et d'autres pour la validation. Dans ce cas, nous ne chercherons pas à visualiser les données mais à extraire d'une population les spectres les plus représentatifs, aussi nous utiliserons des méthodes de sélection de points.

2.1 Étude de fromages par infrarouge

Les produits alimentaires peuvent être considérés comme des systèmes complexes qui présentent différentes propriétés (chimique, microbiologique, etc...) et nécessitent d'être décrits par plusieurs techniques spectroscopiques comme la spectroscopie infrarouge, la spectroscopie par fluorescence, les analyses rhéologiques, les analyses chimiques, ... Aujourd'hui dans l'industrie agroalimentaire, toutes ces mesures conduisent à différentes bases de données effectuées sur les mêmes échantillons. Ces grandes bases de données nécessitent l'utilisation de méthodes permettant de réduire la dimensionnalité afin de visualiser les données tout en conservant un maximum d'informations. En effet, chercher à comprendre un ensemble de données revient à trouver de l'information cachée dans un gros volume de mesures. On peut alors chercher des dépendances linéaires ou non entre les variables pour pouvoir représenter ces dernières dans un espace de plus faible dimension. Nous proposons d'utiliser l'Analyse en Composantes Curvilignes (ACC) et ainsi projeter ce nuage de données non linéairement dépendantes sur une carte de dimension 2.

Les données étudiées [78] sont des spectres infrarouges de fromages mesurant 112 absorbances aux nombres d'onde correspondants (de $1700.9\,cm^{-1}$ à $1486.8\,cm^{-1}$). Les 60 fromages étudiés sont répartis en cinq groupes selon leurs procédés de fabrication. Chaque groupe comporte 12 fromages issus de trois journées de fabrication. Chaque jour, le lait est transformé en 4 fromages, qui présentent des compositions chimiques brutes différentes. Pour chaque groupe, la composition brute de chaque fromage est définie selon un plan factoriel à 2 facteurs à 2 niveaux : la matière sèche et le rapport de la matière grasse sur la matière sèche.

Les 60 fromages sont classés en cinq groupes :

- les pâtes pressées cuites
- les pâtes pressées mi-cuites
- les pâtes pressées
- les pâtes molles
- les pâtes molle stabilisées

A partir des données initiales constituées de 60 spectres et 112 absorbances, nous ne pouvons visualiser que les profils spectraux (figure 2.12) ce qui ne nous permet pas d'identifier les spectres appartenant aux mêmes groupes.

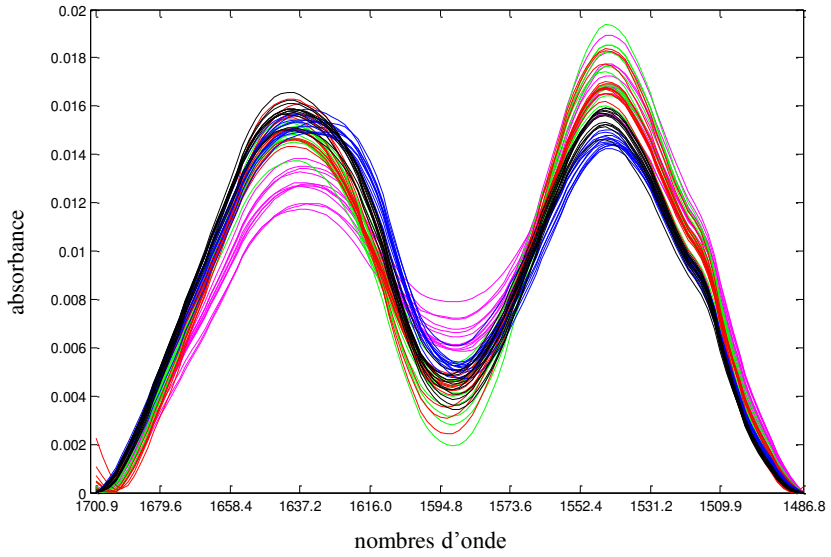


FIGURE 2.12 – Profils spectraux des 60 fromages répartis en cinq groupes. Chaque couleur caractérise un groupe de fromages.

Lorsque nous souhaitons identifier des spectres avec des caractéristiques très proches, nous recommandons d'utiliser l'ACC, puisque les spectres proches dans l'espace initial en grande dimension demeureront proches dans l'espace de projection. Dans cette étude, nous proposons d'appliquer l'ACC sur la base de données sans pré-traitement ce qui nous permet de visualiser les 60 spectres dans un espace bidimensionnel en les regroupant en fonction de leurs caractéristiques. Les résultats sont présentés par la figure 2.13.

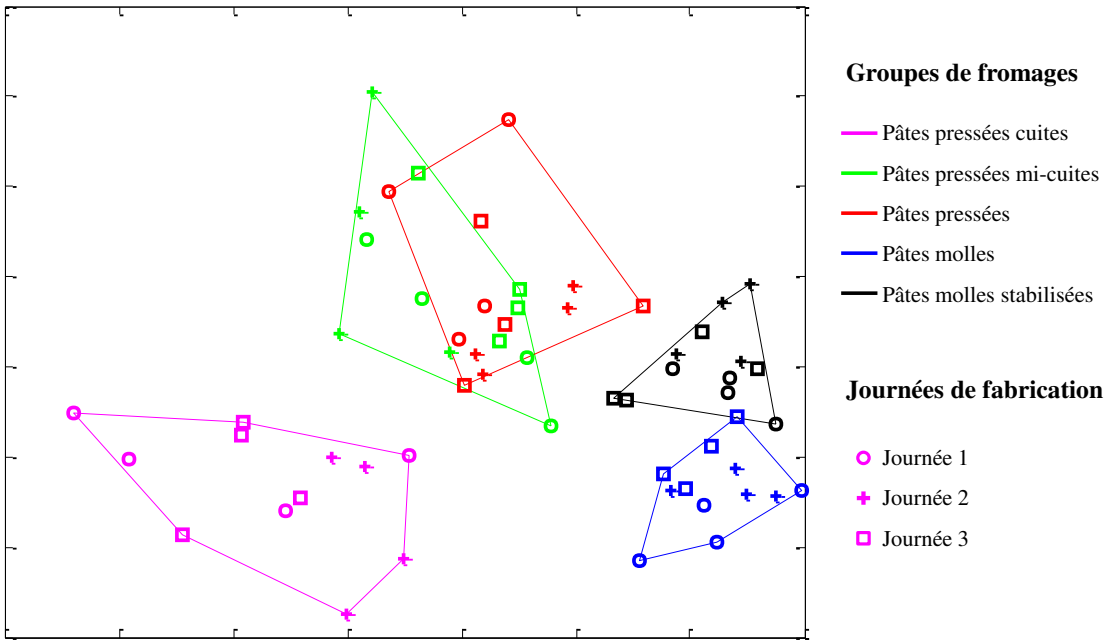


FIGURE 2.13 – Application de l'ACC sur les spectres IR de 60 fromages.

Sur la figure 2.13, nous pouvons différencier quatre groupes correspondants aux pâtes pressées cuites, pâtes molles et pâtes molles stabilisées ; dans le quatrième et dernier groupe il existe une confusion

entre les fromages à pâtes pressées et pâtes pressées mi-cuites qui peut s'expliquer par une composition chimique proche.

A partir de la base de données réelle constituée par les 60 fromages, l'utilisation de l'ACC a permis de réduire la dimensionnalité en projetant les données d'un espace en grande dimension (112 absorbances) vers un espace de plus faible dimension (2 dimensions) tout en conservant au mieux la proximité des points et ainsi l'information locale. Cette étude a permis de visualiser dans une espace en deux dimensions les différents groupes de fromages à partir de leurs spectres infrarouges qui ont été utilisés sans pré-traitement. Toutefois, si nous distinguons nettement des groupes de fromages, l'effet de la journée de fabrication ne peut pas être mis en évidence.

2.2 Étude d'une base de données constituée par des spectres infrarouges

L'objectif de cette étude est de réaliser un échantillonnage approprié qui permettra, avec un nombre réduit de données, de conserver une information de bonne qualité c'est-à-dire la plus représentative possible de l'ensemble initial. Pour cela, nous utiliserons les méthodes de sélection de points que nous avons présentées dans la première partie de ce manuscrit. Nous avons choisi de ne retenir que les algorithmes de Kennard et Stone (KS), WSP et Duplex qui présentent l'avantage de ne pas mettre en jeu un processus stochastique.

Dans le cadre de cette étude, nous disposons de deux bases de données constituées respectivement de 231 et 225 spectres infrarouges mesurant l'absorbance à 800 nombres d'onde pour deux propriétés que nous appellerons "Y1" et "Y2" pour lesquelles nous chercherons à construire des modèles prévisionnels. Nous comparerons les performances des méthodes de sélection, au regard des critères de qualité intrinsèques des sous-ensembles, puis nous effectuerons des modélisations par PLS [76, 77] pour lesquelles nous calculerons et comparerons les critères *a posteriori* tels que l'estimation de l'erreur d'ajustement et de prévision (RMSEC, RMSEP, ...). Enfin, une recherche des outliers (sur les X et/ou sur les Y) sera également réalisée.

2.2.1 Étude des sous-ensembles de calibration et de validation

Les sous-ensembles de calibration et de validation compteront respectivement 80% et 20% des données initiales. L'algorithme KS sera utilisé selon deux approches qui se différencient par la première étape : lorsque l'algorithme de KS est qualifié de "classique", la première étape sélectionne les deux points les plus éloignés, sinon l'algorithme débute par la sélection du point au centre et du point le plus éloigné du centre pour reprendre ensuite la procédure classique de l'algorithme.

L'utilisation des algorithmes de sélection nous conduit à envisager deux stratégies de construction afin de savoir s'il est préférable de répartir uniformément les points dans le set de calibration ou dans le set de validation. Ainsi, la première stratégie consiste à utiliser les algorithmes de sélection pour construire le set de calibration comptant 80% des données initiales et les points restants, c'est-à dire non sélectionnés par les algorithmes, seront affectés au set de validation. La seconde stratégie consiste à construire par algorithme le set de validation comptant 20% des données, les points restants étant affectés au set de calibration. En outre, nous avons choisi d'ajouter un jeu supplémentaire, construit par une sélection aléatoire de 80% des données pour le set de calibration.

Ainsi, nous obtenons 10 jeux de calibration/validation décrits ci-dessous :

- Jeu n°1 : le set de calibration est construit par l'algorithme de KS classique, le set de validation est constitué des 20% de points restants,
- Jeu n°2 : le set de calibration est construit par l'algorithme de KS avec départ du point au centre, le set de validation est constitué des 20% de points restants,
- Jeu n°3 : le set de calibration est construit par l'algorithme WSP, le set de validation est constitué des 20% de points restants,
- Jeu n°4 : le set de validation est construit par l'algorithme de KS classique, le set de calibration est constitué des 80% de points restants,
- Jeu n°5 : le set de validation est construit par l'algorithme de KS avec départ du point au centre, le set de calibration est constitué des 80% de points restants,

- Jeu n°6 : le set de validation est construit par l'algorithme WSP, le set de calibration est constitué des 80% de points restants,
- Jeu n°7 : l'algorithme Duplex est utilisé pour construire en parallèle les sets de calibration et de validation. Ainsi le set de validation est construit par l'algorithme Duplex et le set de calibration est construit pour 20% par Duplex et les points restants,
- Jeux n°8, 9 et 10 : le set de calibration est construit par sélection aléatoire (3 tirages aléatoires), le set de validation est constitué des 20% de points restants.

Pour différencier visuellement les deux stratégies, nous utiliserons un code couleur :

- la couleur (vert, bleu, rouge) foncée caractérise les jeux pour lesquels le set de calibration est construit par algorithme, les points restants constituant le set de validation,
- la couleur (vert, bleu, rouge) claire est utilisée pour caractériser les jeux pour lesquels le set de validation est construit par algorithme, les points restants étant affectés au set de calibration.
- les sous-ensembles construits par l'algorithme Duplex sont représentés en marron,
- les sous-ensembles construits aléatoirement sont présentés en jaune.

La figure 2.14 résume la construction des 10 jeux étudiés.

	Calibration (80%)	Validation (20%)
Jeu n°1	KS classique	Points restants
Jeu n°2	KS point au centre	Points restants
Jeu n°3	WSP	Points restants
Jeu n°4	Points restants	KS classique
Jeu n°5	Points restants	KS point au centre
Jeu n°6	Points restants	WSP
Jeu n°7	Duplex + points restants	Duplex
Jeux n°8 / 9 / 10	Sélection aléatoire	Points restants

FIGURE 2.14 – Construction des 10 jeux de calibration / validation.

Ces méthodes de construction des sous-ensembles de calibration et de validation sont utilisées pour le traitement de deux bases de données. Ainsi, pour un même jeu de données nous obtenons 10 jeux de calibration/validation. Nous comparerons alors la qualité intrinsèque des différents sous-ensembles en termes d'uniformité, par le calcul des critères *Mindist*, *MoyMin* et *Coverage*.

2.2.2 Évaluation de la qualité de modèles de régression PLS

2.2.2.1 Critères *a posteriori* pour comparer les modèles de régression PLS

A partir des différents sous-ensembles nous calculerons des modèles de régression PLS. Pour les valider, nous proposons de calculer les critères *a posteriori* suivants :

- L'erreur quadratique moyenne MSE (Mean Squared Error) (équation (2.1)) ainsi que sa racine carrée (RMSE) (équation (2.2)) définies respectivement par :

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.1}$$

$$RMSE = \sqrt{MSE} \tag{2.2}$$

avec y_i la réponse aux points considérés, $\hat{y}_i$ la réponse calculée par le modèle de régression PLS aux points considérés et N le nombre de points considérés.

– Le coefficient de détermination R^2 (équation (2.3)) qui mesure la qualité de l'ajustement défini par :

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.3}$$

avec $\bar{y}$ la valeur moyenne de la réponse.

– L'erreur absolue maximale définie par (équation (2.4)) :

$$MAX = max|y_i - \hat{y}_i| \tag{2.4}$$

Les critères présentés ci-dessus seront calculés pour le set de calibration, notés MSEC, RMSEC, R²cal, MAXcal, et pour le set de validation notés MSEP, RMSEP, MAXval. Un modèle de bonne qualité sera caractérisé par de faibles valeurs de RMSEC, RMSEP et MAX, et un coefficient de détermination R^2 proche de 1.

2.2.2.2 Critères pour la détection des outliers

– Pour la recherche des outliers spectraux, nous considèrerons le résidu spectral noté *Qres* et le critère *leverage* :

- Le résidu spectral Qres est défini par (équation (2.5)) :

$$Qres = e_i.e_i^T \tag{2.5}$$

où $e_i = X_{c_i} - t_i.P^T$ avec X_{c_i} le spectre centré, t_i le score de l'individu i dans le modèle, P la matrice des loadings pour les k composantes du modèle.

- Le critère *leverage* est défini par (équation (2.6)) :

$$h_i = \sum_{l=1}^{k}\left(\frac{t_{il}^2}{t_l^T.t_i}\right) \tag{2.6}$$

avec t_{il} le score de l'individu i pour la composante l dans le modèle, k le nombre de composantes dans le modèle.

Un outlier spectral sera caractérisé par des valeurs élevées de *Qres* et/ou *leverage*.

– Pour la recherche des outliers en Y nous calculons les résidus (équation (2.7)) entre la réponse étudiée et la réponse calculée par le modèle de régression PLS :

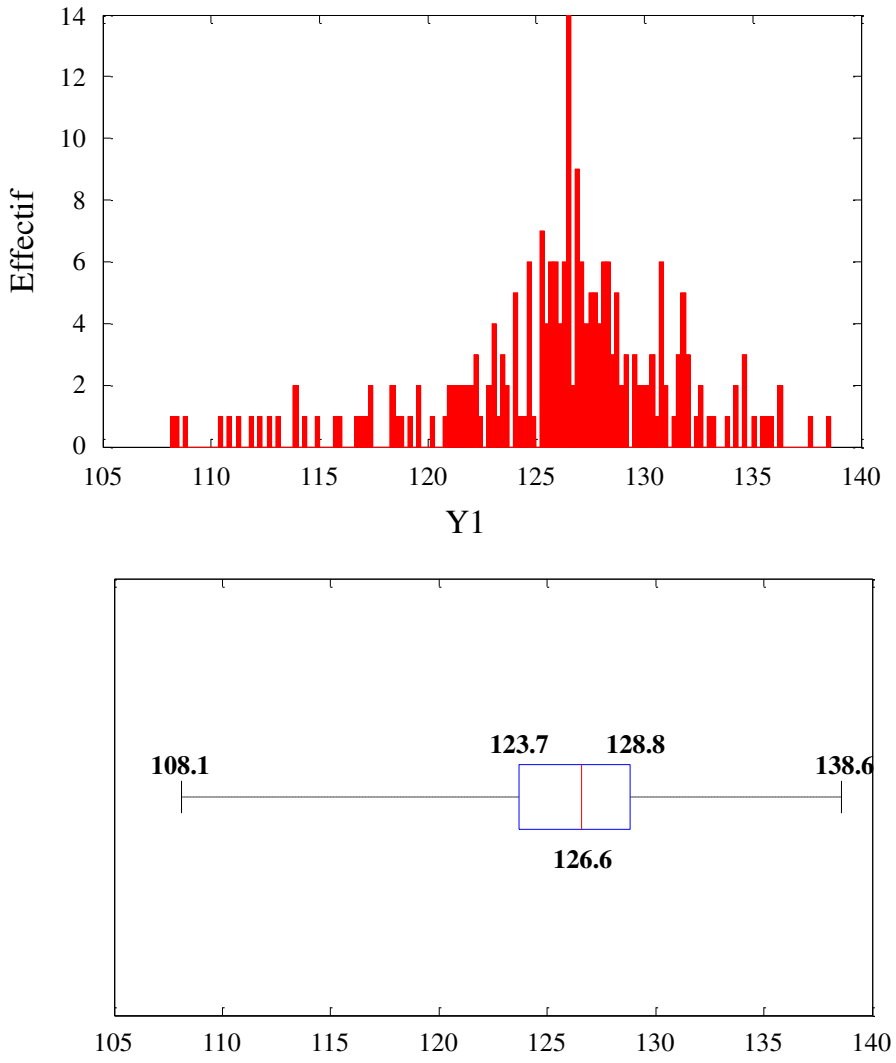
$$e_i = y_i - \hat{y}_i \tag{2.7}$$

avec y_i la réponse aux points considérés, $\hat{y}_i$ la réponse calculée par le modèle de régression PLS aux points considérés.

2.2.3 Étude de la réponse Y1

Afin de tester et comparer les méthodes de sélection, nous avons choisi d'étudier la réponse "Y1" mesurée pour la première base de données avec 231 spectres acquis pour 800 nombres d'onde.

Dans un premier temps, nous pouvons utiliser les statistiques descriptives qui permettent de visualiser la distribution de la réponse "Y1" sous la forme d'un histogramme des effectifs et d'un box plot [61] (figure 2.15). Nous observons que la répartition de la réponse est comprise entre 108.1 et 138.6.



Réponse	Moyenne	Écart-Type	Minimum	Maximum	Q1	Q2 (médiane)	Q3
Y1	125.94	5.47	108.1	138.6	123.7	126.6	128.8

FIGURE 2.15 – Représentations graphiques de la répartition de la réponse "Y1".

2.2.3.1 Construction et caractérisation des sous-ensembles de calibration et de validation

Pour la réponse "Y1", nous rappelons que nous disposons d'un jeu de données constitué de 231 spectres ce qui implique de construire un sous-ensemble de calibration à 185 points et un sous-ensemble de validation à 46 points selon les différentes stratégies regroupées dans le figure 2.14. Les figures 2.16 et 2.17 présentent respectivement les valeurs des critères *Mindist* et *Coverage* des sous-ensembles de calibration et de validation.

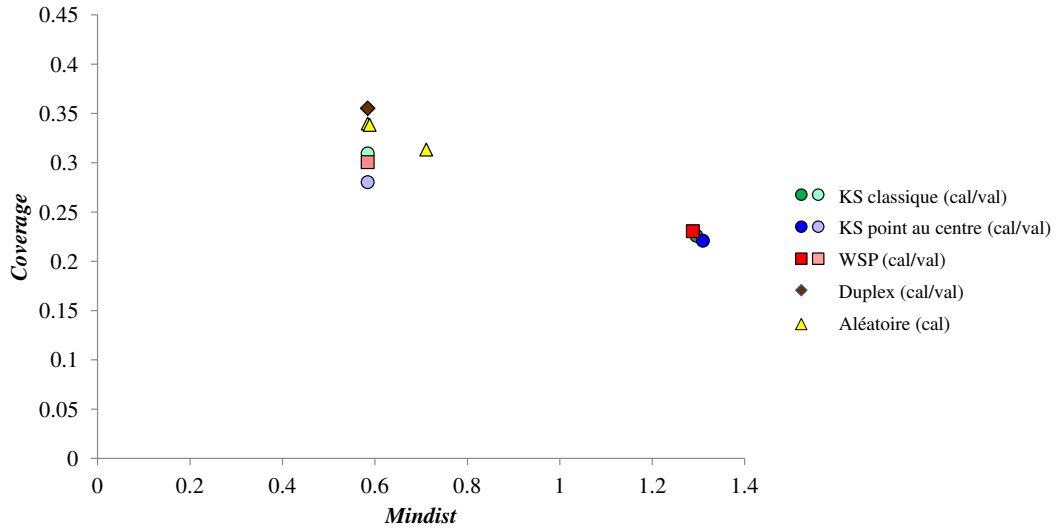


FIGURE 2.16 – Comparaison de la qualité des **sous-ensembles de calibration** contenant 80% des données initiales ($N = 185$ points). Pour différencier les deux stratégies de construction, nous utilisons un code couleur. La couleur foncée caractérise les jeux pour lesquels le set de calibration est construit par algorithme, les points restants constituant le set de validation. La couleur claire est utilisée pour caractériser les jeux pour lesquels le set de validation est construit par algorithme, les points restants étant affectés au set de calibration. Certains sous-ensembles n'apparaissent pas sur ce graphe car ils présentent des valeurs très similaires à d'autres sets de calibration.

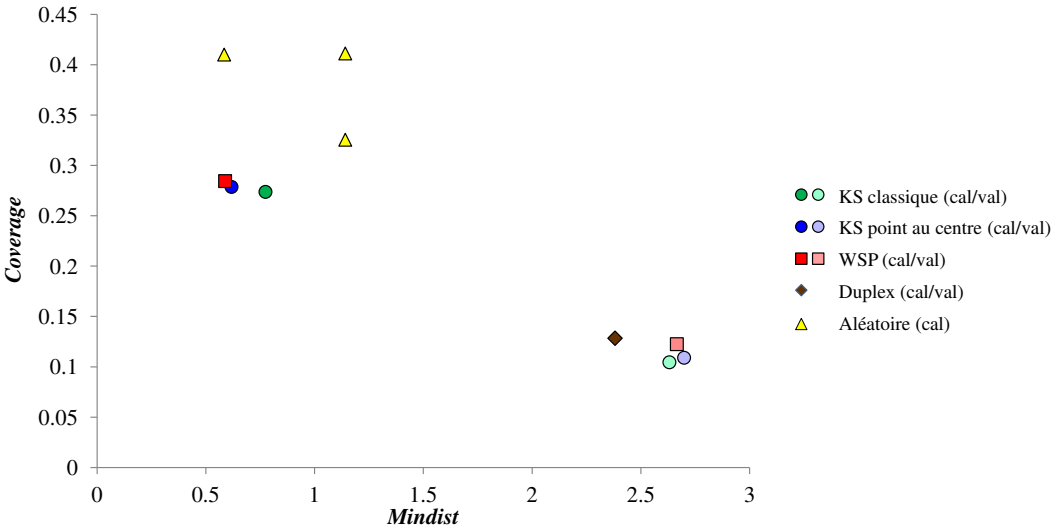


FIGURE 2.17 – Comparaison de la qualité des **sous-ensembles de validation** contenant 20% des données initiales ($N = 46$ points).

Les figures 2.16 et 2.17 mettent en exergue deux groupes. En effet, quel que soit le sous-ensemble étudié, calibration ou validation, le groupe de points qui présente les meilleurs critères correspond systématiquement aux sous-ensembles qui ont été sélectionnés par un algorithme de sélection et non ceux constitués des points restants. Les sous-ensembles construits aléatoirement présentent une qualité qui varie en fonction du tirage et les critères diffèrent de ceux des sous-ensembles construits par un algorithme de sélection.

2.2.3.2 Détermination du nombre de composantes pour la modélisation par PLS

Toutes les régressions PLS qui sont présentées dans ce manuscrit ont été effectuées avec la fonction 'plsregress' du logiciel Matlab [79].

Dans cet exemple, nous utilisons la régression PLS pour analyser la réponse "Y1". Nous considérons un ensemble de calibration constitué par 185 points (80%) et un ensemble de validation comptant 46 points (20%). Afin de construire le modèle de régression PLS avec un nombre satisfaisant de composantes, nous proposons dans un premier temps de réaliser la régression PLS avec 15 composantes. A partir de cette première régression, nous pouvons déterminer le pourcentage de la variance de la réponse "Y1" expliqué par composante PLS (figure 2.18) et calculer l'évolution du critère MSEC en fonction du nombre de composantes (figure 2.19), ce qui nous permettra de choisir le nombre de composantes PLS à retenir.

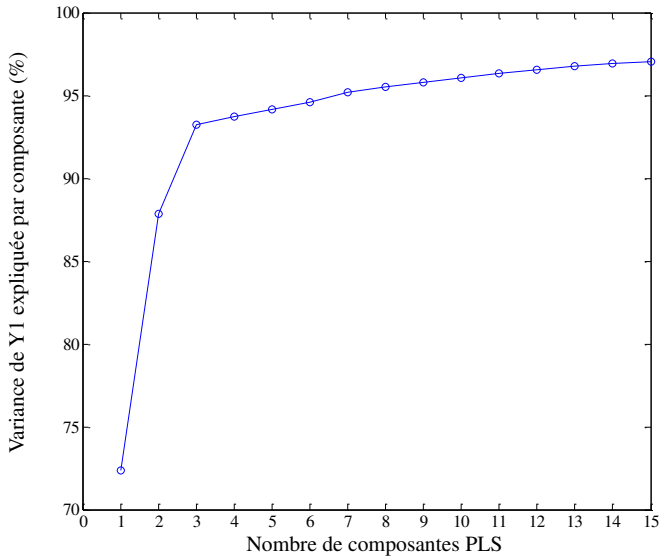


FIGURE 2.18 – Représentation du pourcentage de variance cumulée – Réponse "Y1".

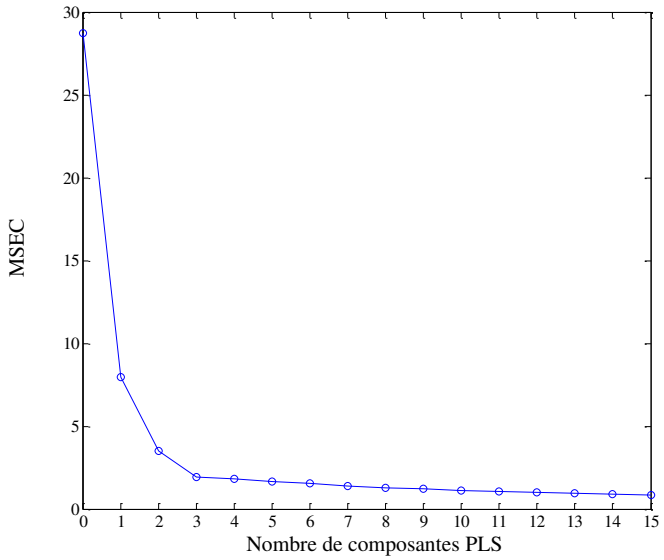


FIGURE 2.19 – Représentation du MSEC en fonction du nombre de composantes PLS – Réponse "Y1".

Sur les figures 2.18 et 2.19, nous observons un MSEC stable à partir de 4 composantes qui expliquent 94% de la variance de la réponse "Y1". Ainsi, pour la suite de notre étude nous choisissons de retenir 4 composantes pour la régression PLS.

2.2.3.3 Calcul des critères *a posteriori*

Afin de comparer les différentes stratégies, nous proposons de calculer les critères *a posteriori* des sous-ensembles de calibration constitués par 185 spectres (80%) et de validation comptant 46 points (20%). Les valeurs de ces critères sont reportées dans le tableau 2.8.

Tableau 2.8 – Tableau récapitulatif des critères *a posteriori* calculés sur la première base de données pour l'analyse de la réponse "Y1".

CALIBRATION (185 points)	VALIDATION (46 points)	RMSEC	R²cal	MAXcal	RMSEP	MAXval
Jeu n°1		1.2451	0.946	5.477	1.4984	4.6061
Jeu n°2		1.2797	0.9437	5.6047	1.3577	4.3968
Jeu n°3		1.2614	0.9446	5.4221	1.4025	4.6399
Jeu n°4		1.3135	0.9295	5.3074	1.3041	2.9923
Jeu n°5		1.275	0.9333	5.199	1.498	3.9896
Jeu n°6		1.29	0.9338	5.4121	1.3213	4.0778
Jeu n°7		1.2818	0.9408	5.4342	1.3825	4.2157
Jeu n°8		1.27	0.9508	5.5715	1.5258	4.514
Jeu n°9		1.2591	0.9477	5.5721	1.5295	3.8489
Jeu n°10		1.2475	0.9477	5.5742	1.6016	4.3839

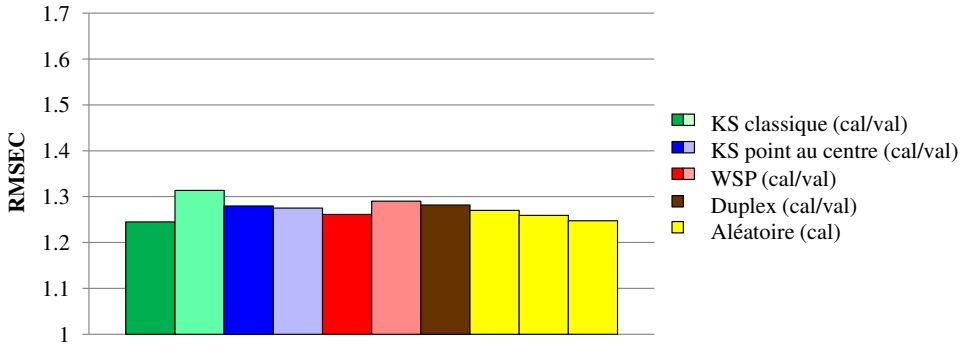


FIGURE 2.20 – RMSEC obtenus à partir des sous-ensembles de calibration pour l'analyse de la réponse "Y1".

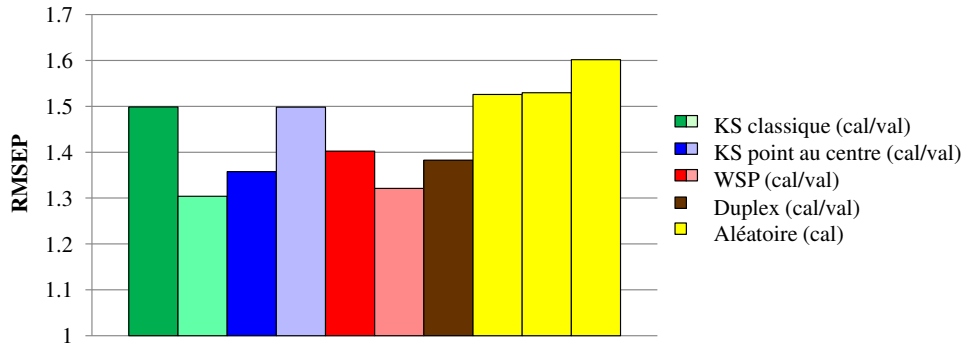


FIGURE 2.21 – RMSEP obtenus à partir des sous-ensembles de validation pour l'analyse de la réponse "Y1".

On observe sur les figures 2.20 et 2.21 des valeurs de RMSEC proches de 1.2 quelle que soit la méthode de sélection utilisée pour construire le sous-ensemble de calibration et des valeurs RMSEP qui fluctuent légèrement autour de 1.3. De manière générale, un modèle sera considéré comme performant lorsque les valeurs RMSEC et RMSEP sont équivalentes.

D'autre part, nous pouvons étudier le coefficient de détermination R^2 :

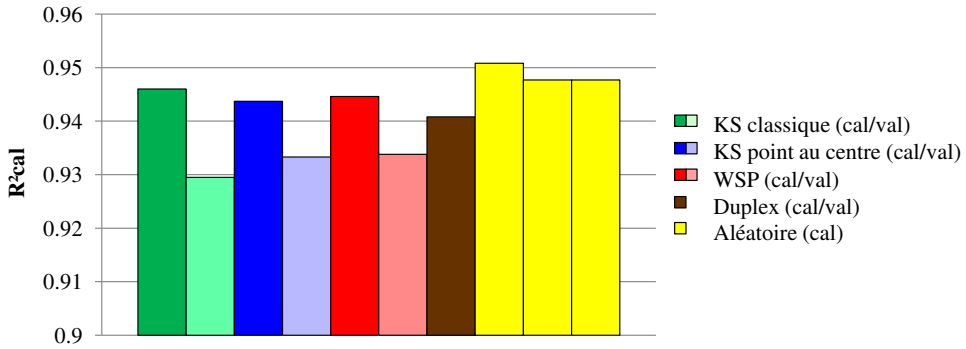


FIGURE 2.22 – R^2 calculés à partir des sous-ensembles de calibration pour l'analyse de la réponse "Y1".

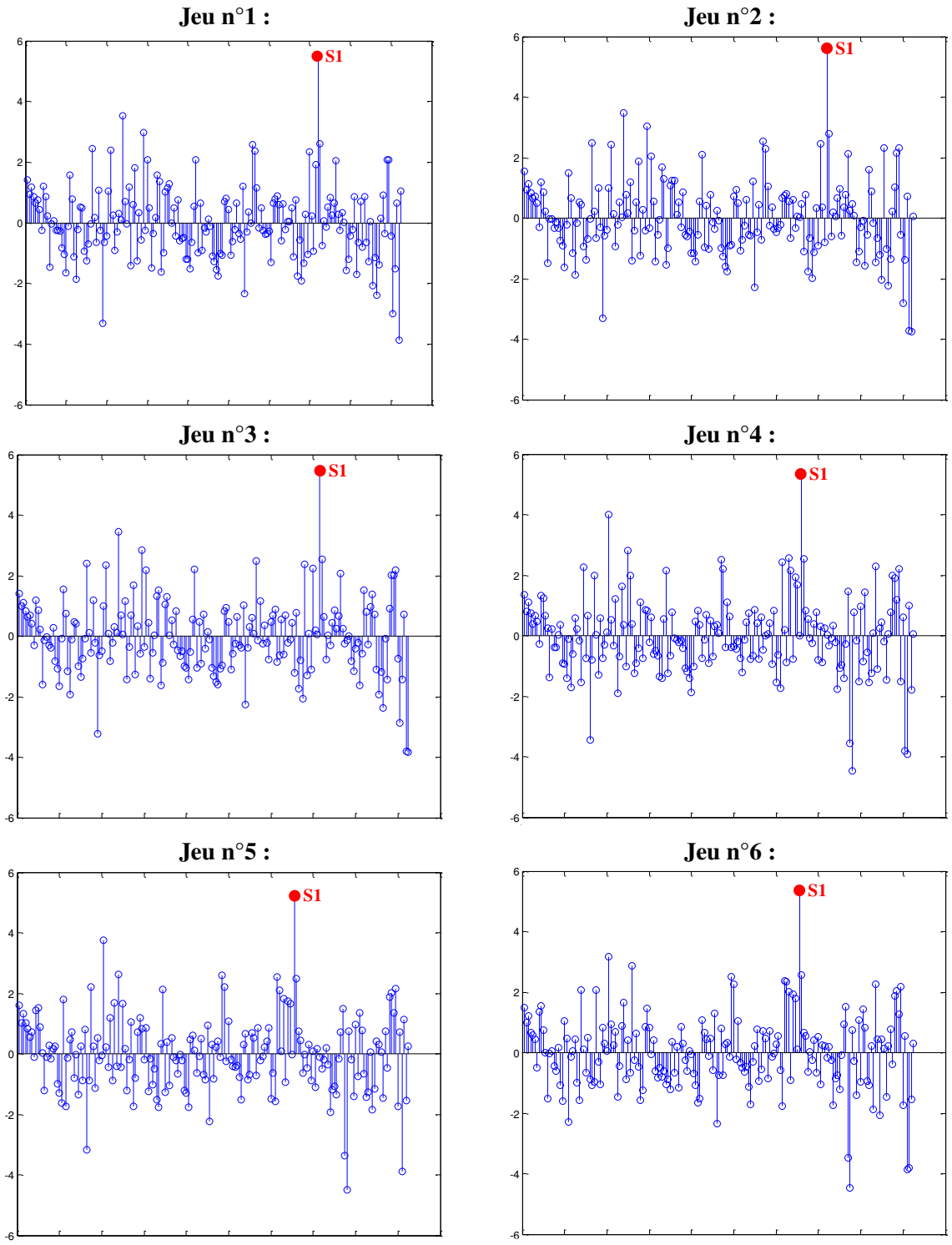
Lors de l'analyse du critère R^2 (figure 2.22) les valeurs sont proches et présentent peu de variation mais avec une légère préférence pour la construction des sets de calibration par algorithme. Nous rappelons que cette première régression PLS prend en compte tous les points constituant le set de calibration.

2.2.3.4 Analyse des résidus en Y

Dans un premier temps nous nous intéressons aux résidus en Y qui se définissent comme la différence entre la réponse étudiée et la réponse calculée par le modèle de régression PLS (équation (2.8)).

$$e_i(Y1) = Y1_i - Y1_{i-calculé} \tag{2.8}$$

Nous proposons de représenter sur la figure 2.23 les valeurs de ces résidus en Y obtenues pour chaque stratégie de construction des sous-ensembles de calibration et un point sera qualifié d'outlier en Y s'il présente une valeur absolue de résidu beaucoup plus grande que celle des autres points.



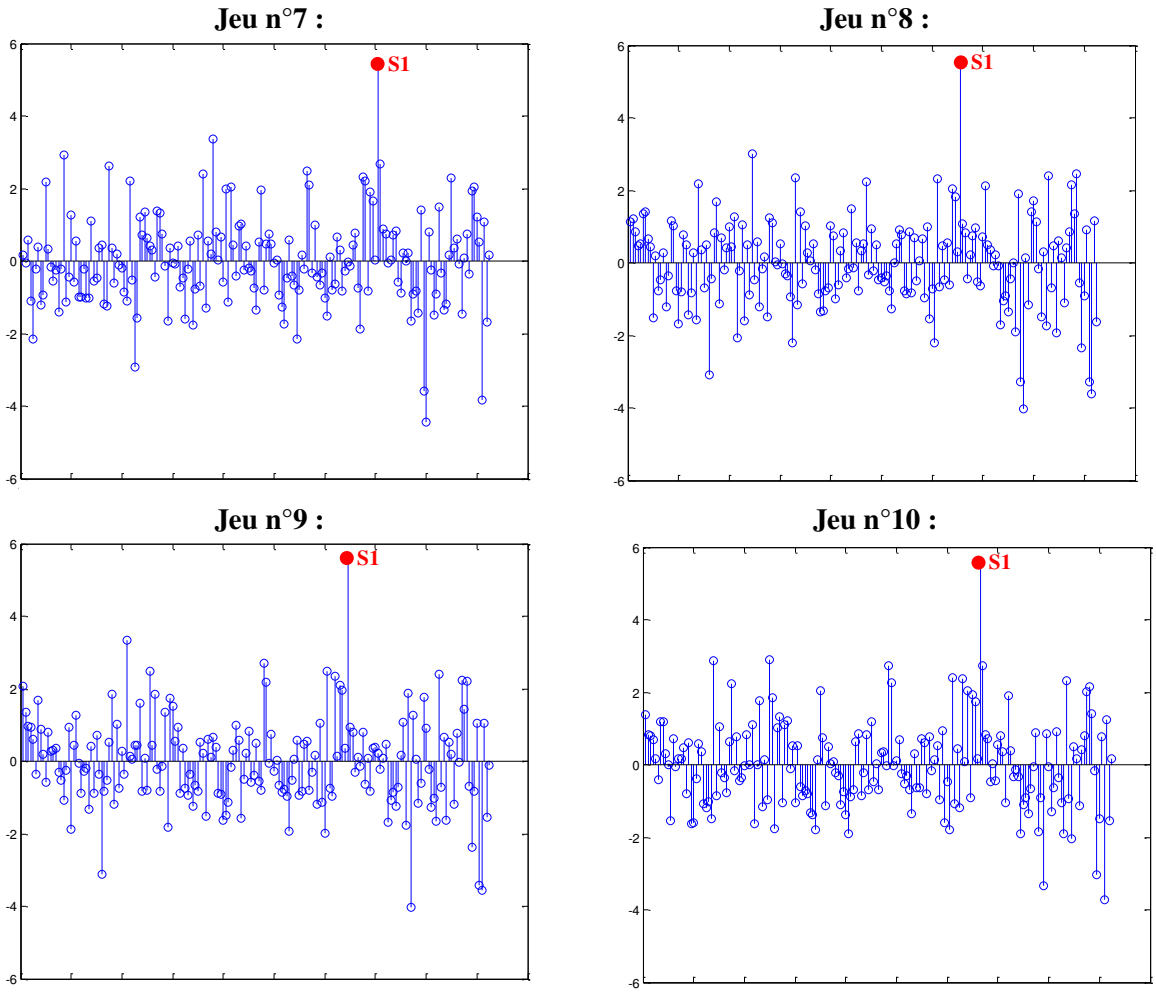


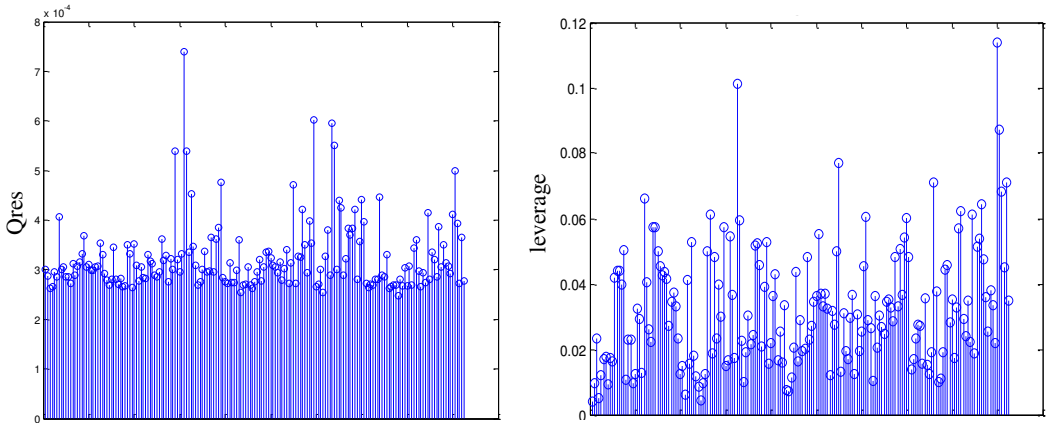
FIGURE 2.23 – Résidus en Y obtenus pour l'analyse de la réponse "Y1".

Sur la figure 2.23, nous observons que le spectre **S1** présente systématiquement une valeur absolue de résidu supérieure aux autres et ce quelle que soit la stratégie de construction du set de calibration.

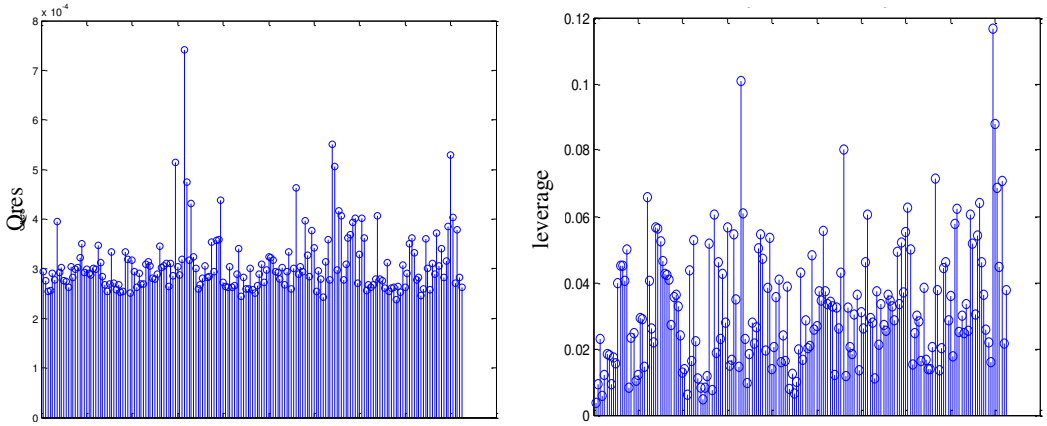
2.2.3.5 Analyse des résidus en X

Après avoir étudié la présence d'outliers en Y, l'étude est complétée par la recherche d'outliers spectraux (figure 2.24) qui seront caractérisés par des valeurs élevées des critères *Qres* et/ou *leverage* par rapport aux autres spectres constituant le set de calibration.

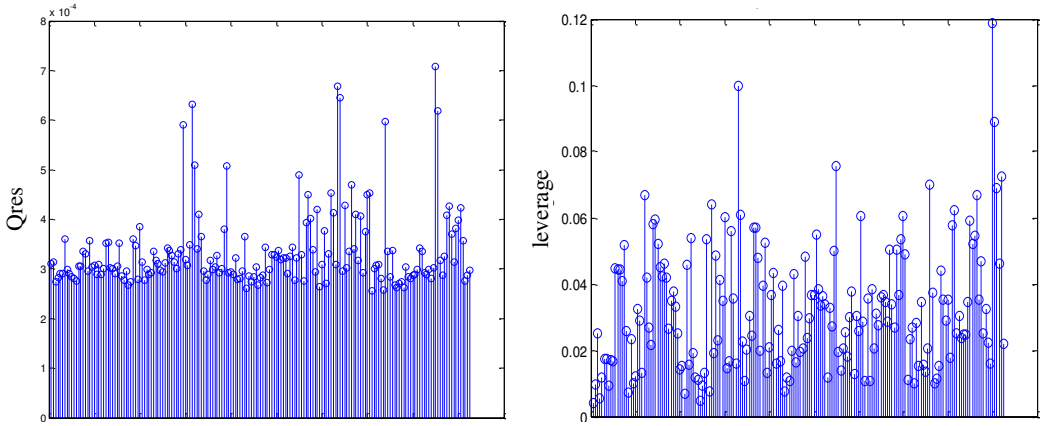
Jeu n°1 :



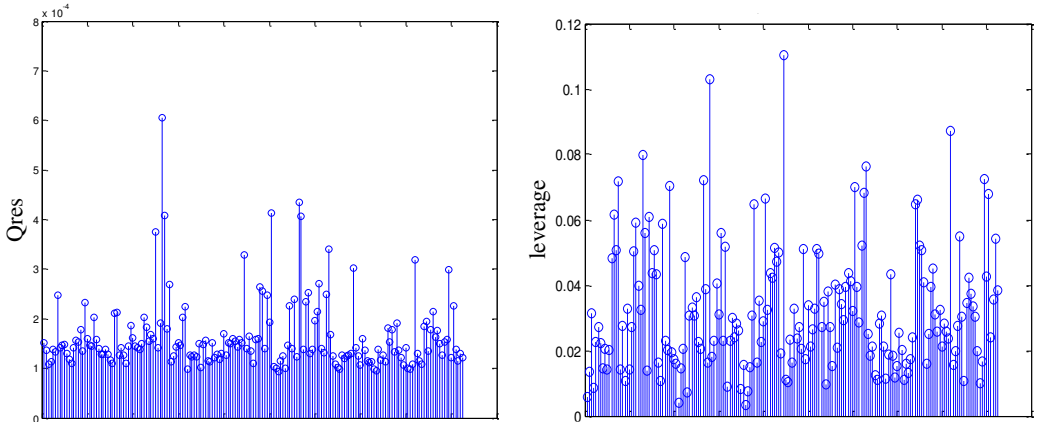
Jeu n°2 :



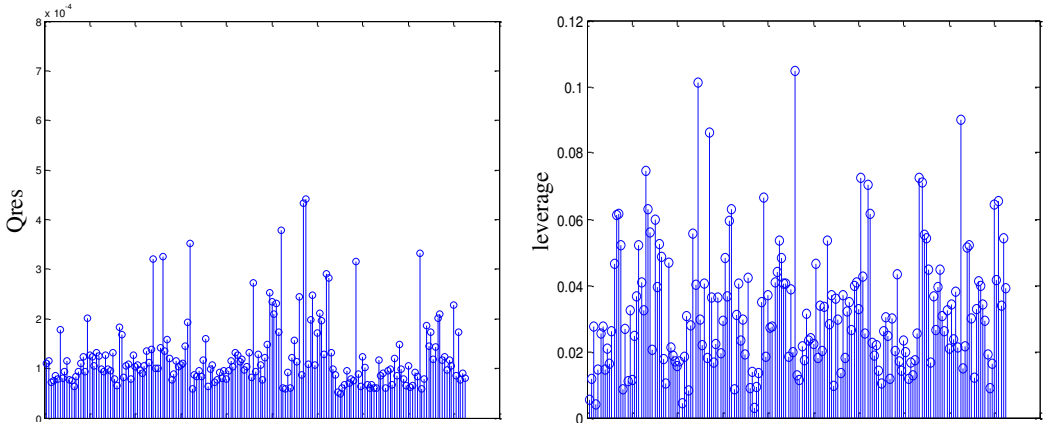
Jeu n°3 :



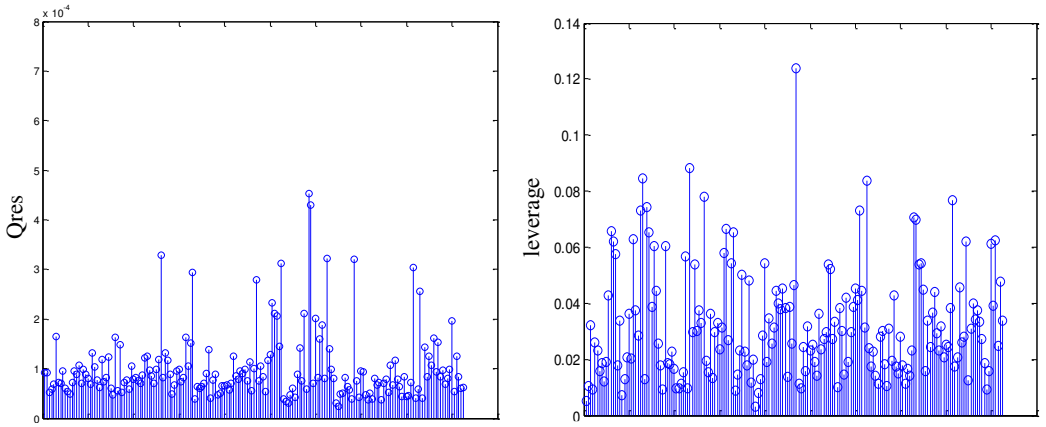
Jeu n°4 :



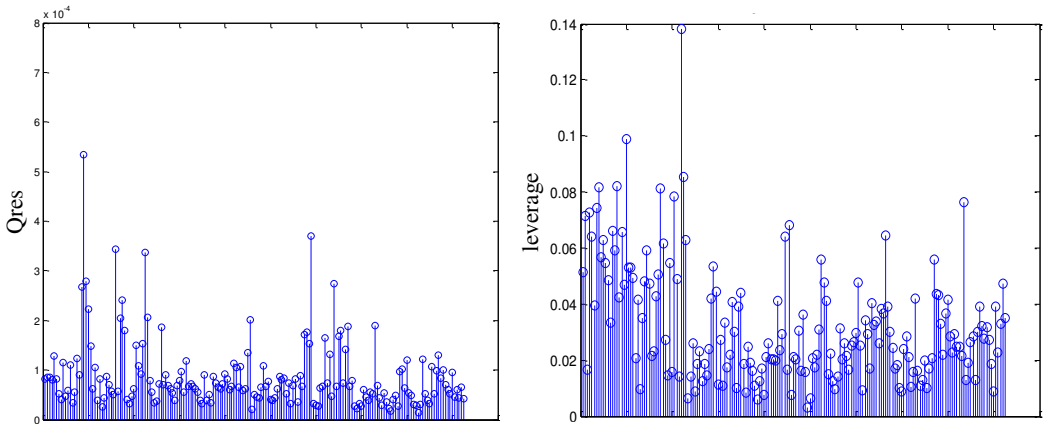
Jeu n°5 :



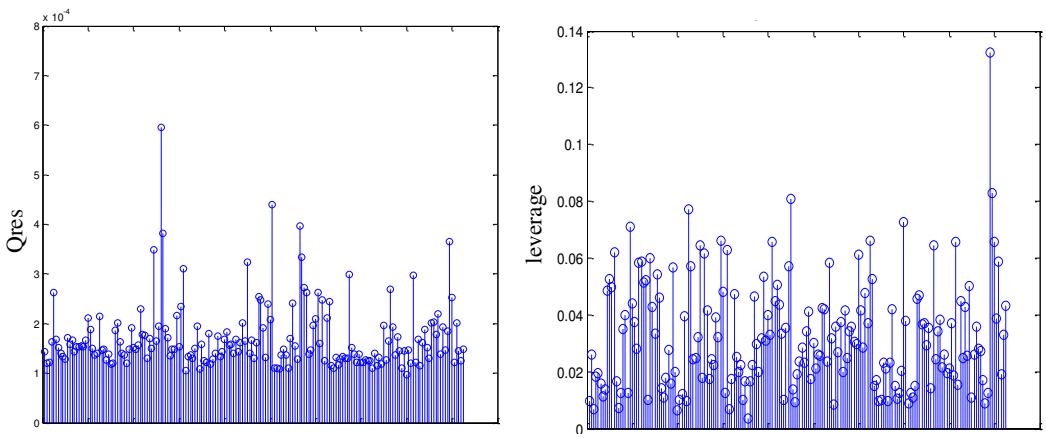
Jeu n°6 :



Jeu n°7 :



Jeu n°8 :



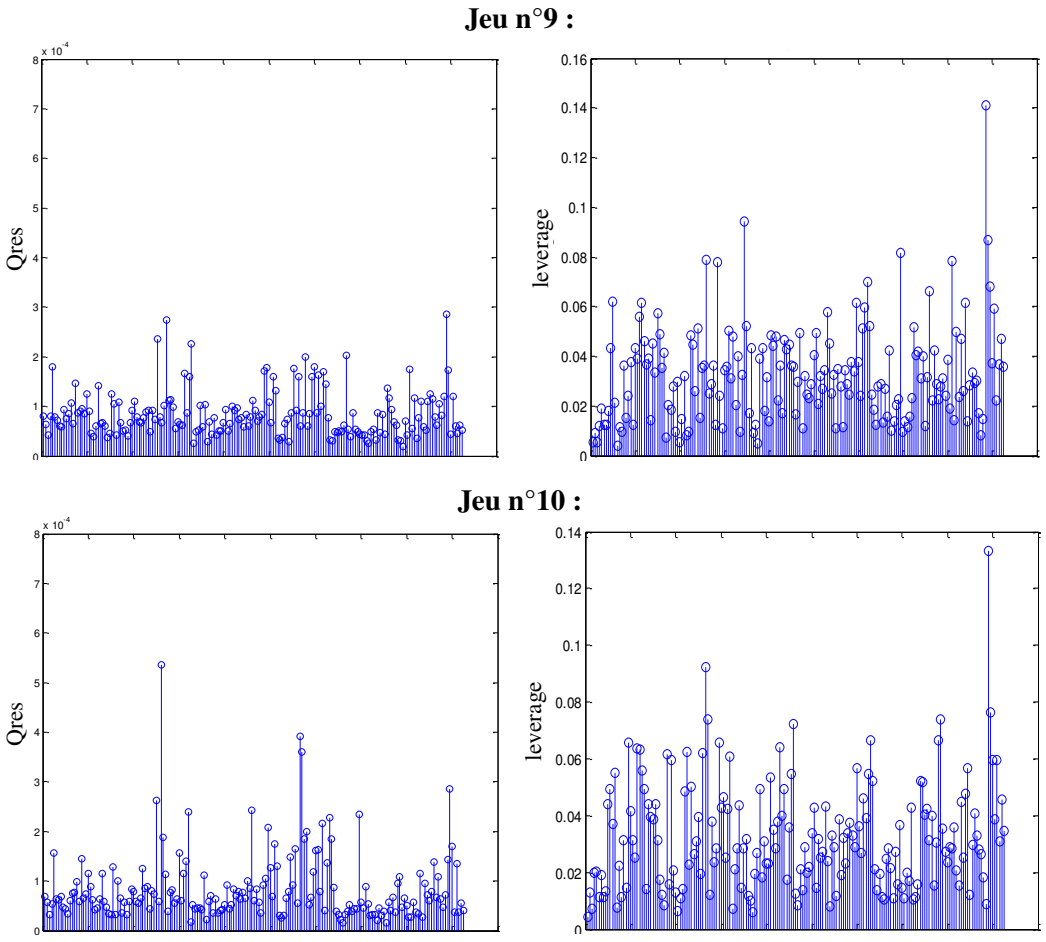


FIGURE 2.24 – Résidus en X obtenus pour l'analyse de la réponse "Y1".

Sur la figure 2.24, il est difficile d'identifier un outlier spectral. En effet, les faibles valeurs des critères *Qres* et *leverage* ne conduisent pas à des différences significatives entre les différents spectres du set de calibration. Toutefois, si ces valeurs sont acceptables, nous pouvons observer des spectres qui se distinguent légèrement des autres.

2.2.3.6 Calcul des critères *a posteriori* après suppression des outliers

Afin de vérifier si la qualité des modèles est améliorée après la suppression du spectre **S1** (détecté comme outlier en Y) du set de calibration, nous proposons de recalculer les critères *a posteriori* des sous-ensembles de calibration et de validation (figures 2.25 à 2.27). Les valeurs de ces critères sont reportées dans le tableau 2.9.

Tableau 2.9 – Tableau récapitulatif des critères *a posteriori* calculés pour l'analyse de la réponse "Y1" après suppression du spectre **S1** identifié comme outlier en Y.

CALIBRATION (184 points)	VALIDATION (46 points)	RMSEC	R²cal	MAXcal	RMSEP	MAXval
Jeu n°1		1.185	0.9511	3.9389	1.5027	4.5573
Jeu n°2		1.2189	0.9489	3.7866	1.3792	4.3386
Jeu n°3		1.2052	0.9495	3.8854	1.3999	4.5987
Jeu n°4		1.2591	0.935	4.4585	1.2942	3.1789
Jeu n°5		1.2227	0.9384	4.4753	1.4807	4.048
Jeu n°6		1.2358	0.9391	4.4463	1.3292	4.0007
Jeu n°7		1.2238	0.946	4.4325	1.381	4.1145
Jeu n°8		1.2077	0.9555	3.9816	1.5461	4.5701
Jeu n°9		1.2032	0.9522	4.0197	1.5449	3.9008
Jeu n°10		1.1822	0.953	3.763	1.6222	4.3264

La comparaison des critères *a posteriori* regroupés dans les tableaux 2.8 et 2.9 permet de constater que la suppression du spectres S1 n'améliore pas significativement la qualité du modèle.

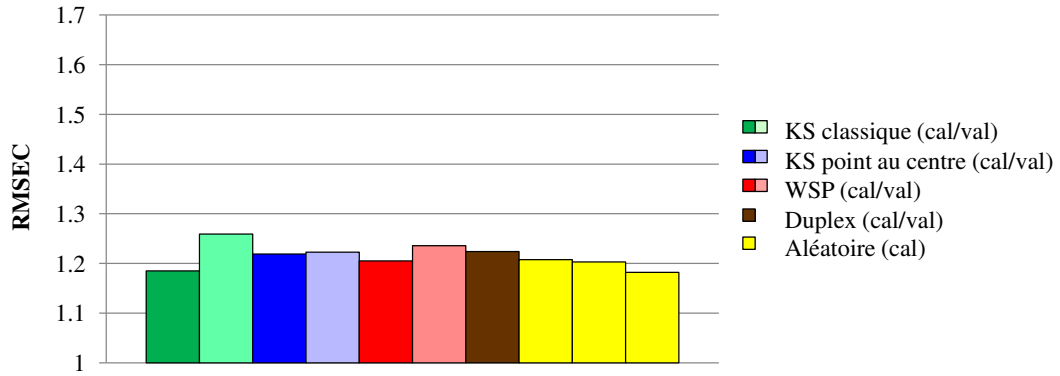


FIGURE 2.25 – RMSEC obtenus à partir des sous-ensembles de calibration sans le spectre **S1** pour l'analyse de la réponse "Y1".

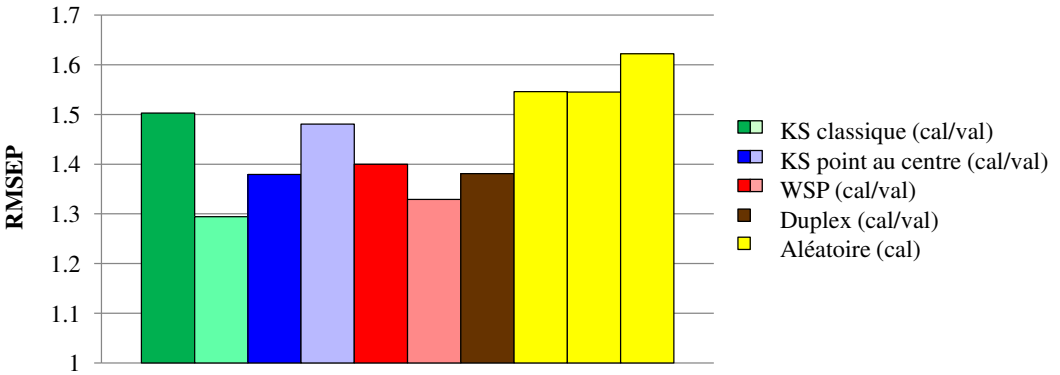


FIGURE 2.26 – RMSEP obtenus à partir des sous-ensembles de validation sans le spectre **S1** pour l'analyse de la réponse "Y1".

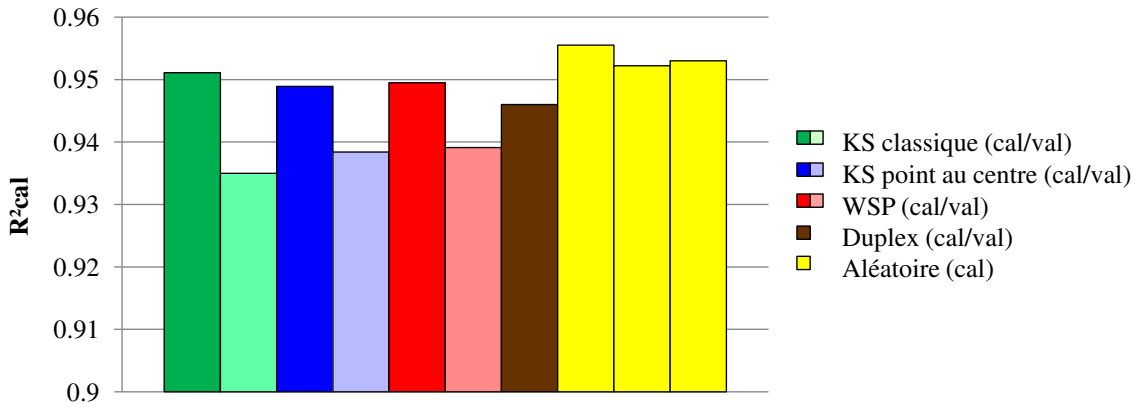
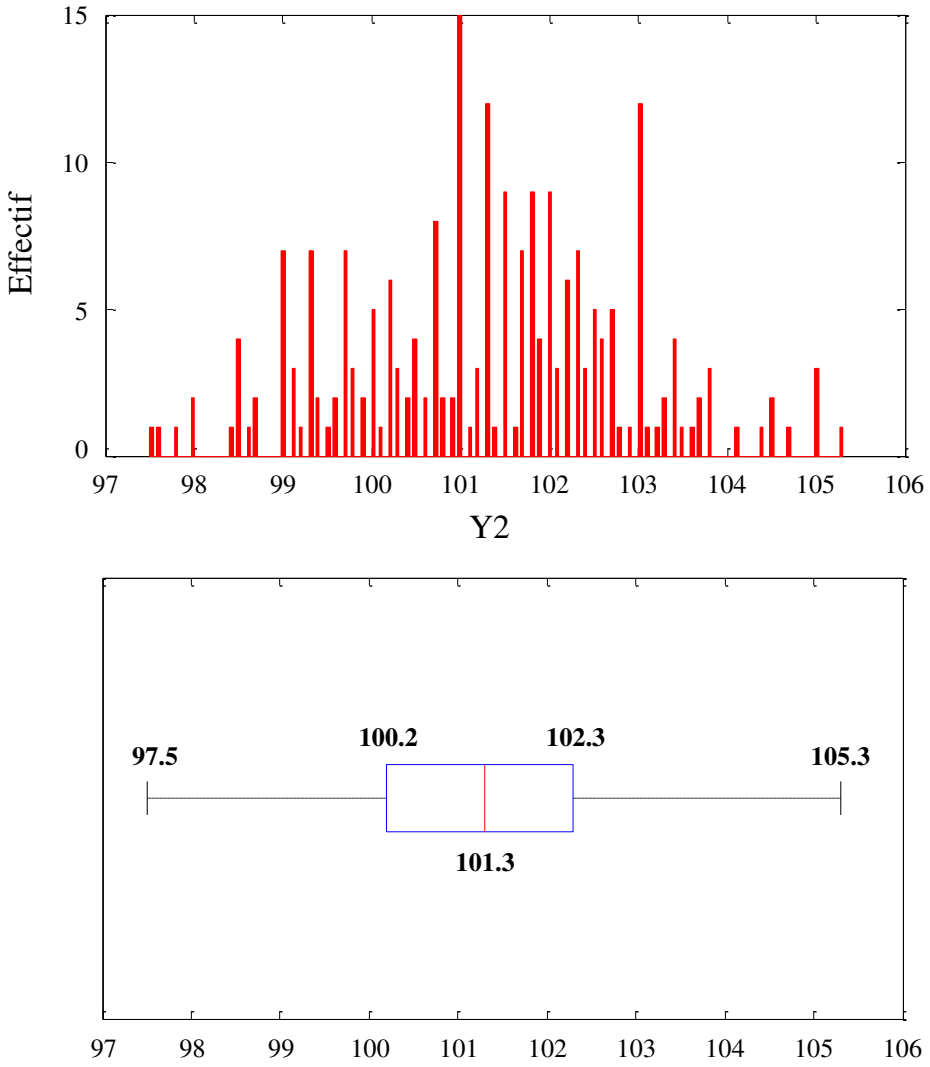


FIGURE 2.27 – R^2 calculés à partir des sous-ensembles de calibration sans le spectre **S1** pour l'analyse de la réponse "Y1".

Si nous comparons le valeurs des critères *a posteriori* avant et après la suppression du spectre **S1**, nous observons une légère amélioration de ces critères lorsque nous le supprimons des sous-ensembles de calibration. Cependant, il reste difficile de conclure sur l'efficacité d'une méthode de construction par rapport à une autre même s'il semble préférable de construire le set de calibration par algorithme.

2.2.4 Étude de la réponse Y2

Les résultats de l'étude précédente ne conduisent pas à une recommandation universelle pour le choix de la méthode de sélection. Ces observations nous ont donc conduits à reconsidérer la démarche, quant au choix des points. En effet, la suppression des outliers d'un ensemble de calibration crée probablement des lacunes dans l'espace et modifie alors le bon conditionnement de cet ensemble de points qui est destiné à l'apprentissage. Pour pallier cette défaillance, nous proposons une nouvelle démarche en reconstruisant de nouveaux sets de calibration après suppression des outliers en Y. Un histogramme des effectifs et un graphe box plot (figure 2.28) permettent de visualiser la répartition des valeurs de la réponse "Y2" : nous observons une dispersion des valeurs de 97.5 à 105.3.



Réponse	Moyenne	Écart-Type	Minimum	Maximum	Q1	Q2 (médiane)	Q3
Y2	101.30	1.57	97.5	105.3	100.2	101.3	102.3

FIGURE 2.28 – Représentations graphiques de la répartition de la réponse "Y2".

2.2.4.1 Construction et caractérisation des sous-ensembles de calibration et de validation

Pour construire les sous-ensembles, nous utilisons les mêmes stratégies que celles présentées dans la section 2.2.1, permettant d'obtenir 10 jeux de calibration/validation en respectant les mêmes

proportions de points 80% / 20%. Les figures 2.29 et 2.30 comparent les critères d'uniformité des sous-ensembles de calibration et de validation.

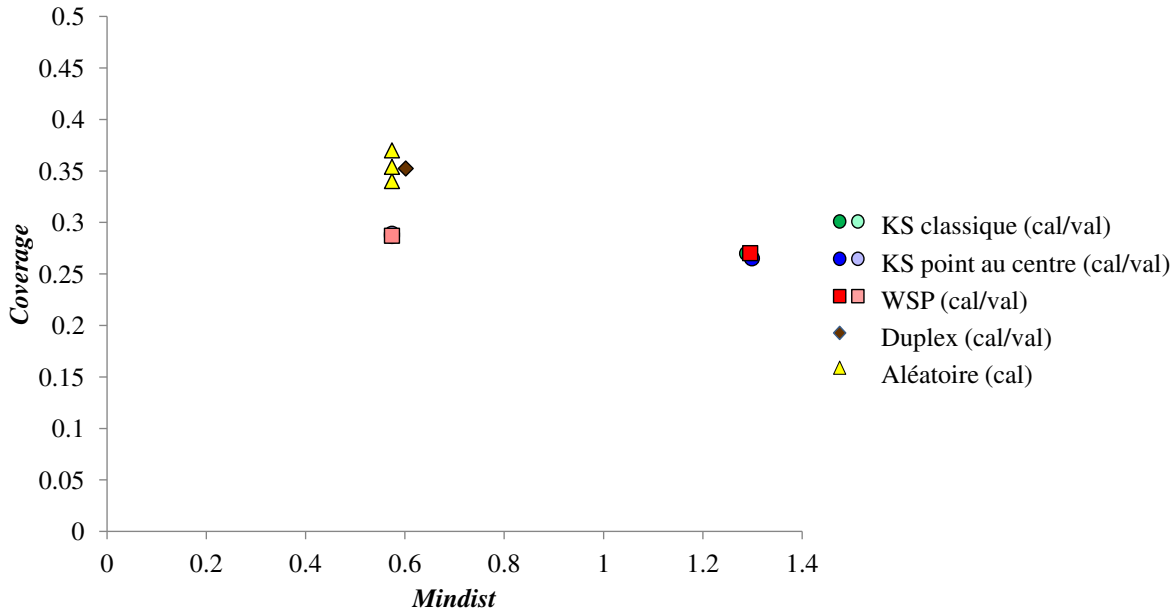


FIGURE 2.29 – Comparaison de la qualité intrinsèque des **sous-ensembles de calibration** contenant 80% des données initiales ($N = 180$ points). Certains sous-ensembles n'apparaissent pas sur ce graphe car ils présentent des valeurs très similaires à d'autres sets de calibration.

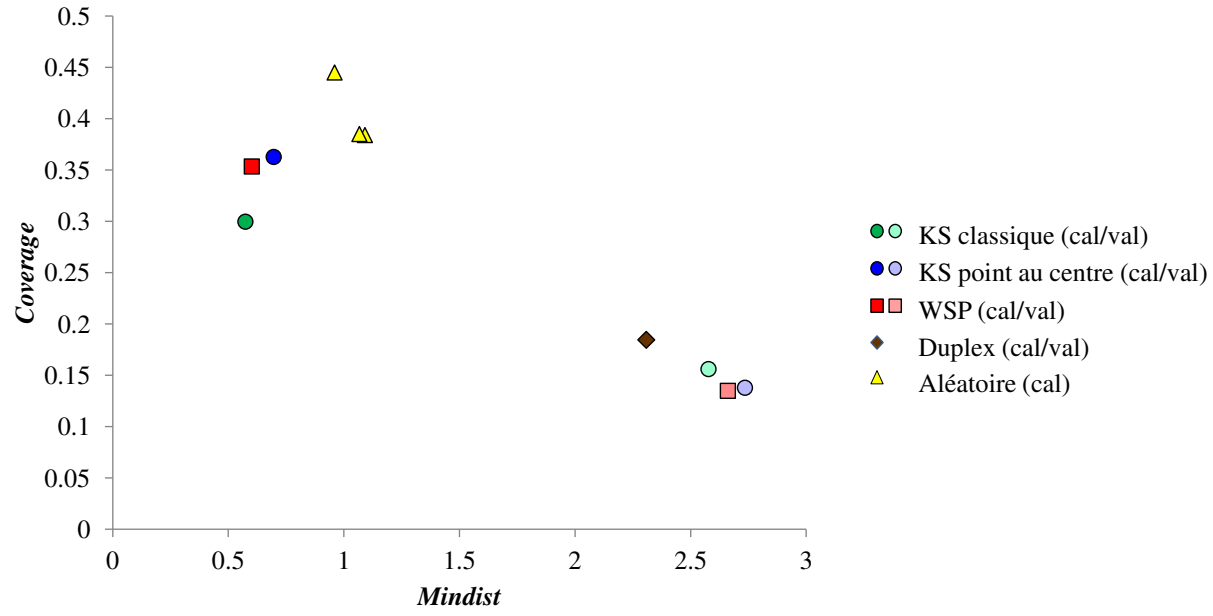


FIGURE 2.30 – Comparaison de la qualité intrinsèque des **sous-ensembles de validation** contenant 20% des données initiales ($N = 45$ points).

Les figures 2.29 et 2.30 conduisent à la même conclusion que la première base de données. En effet, le sous-ensemble étudié sera de meilleure qualité lorsque ce dernier est construit par un algorithme de sélection et non par les points restants.

2.2.4.2 Détermination du nombre de composantes pour la modélisation par PLS

Pour chaque sous-ensemble, nous souhaitons construire un modèle de régression PLS qui nécessite de définir au préalable le nombre de composantes PLS à retenir. Pour ce faire, nous considérons un ensemble de calibration constitué par 180 points (80%) et un ensemble de validation comptant 45 points (20%). Afin de construire le modèle de régression PLS avec un nombre satisfaisant de composantes, nous proposons dans un premier temps de réaliser la régression PLS avec 15 composantes, ce qui nous permet de déterminer le pourcentage de la variance de la réponse "Y2" expliqué par composante PLS (figure 2.31) et de calculer l'évolution du critère MSEC en fonction du nombre de composantes PLS (figure 2.32).

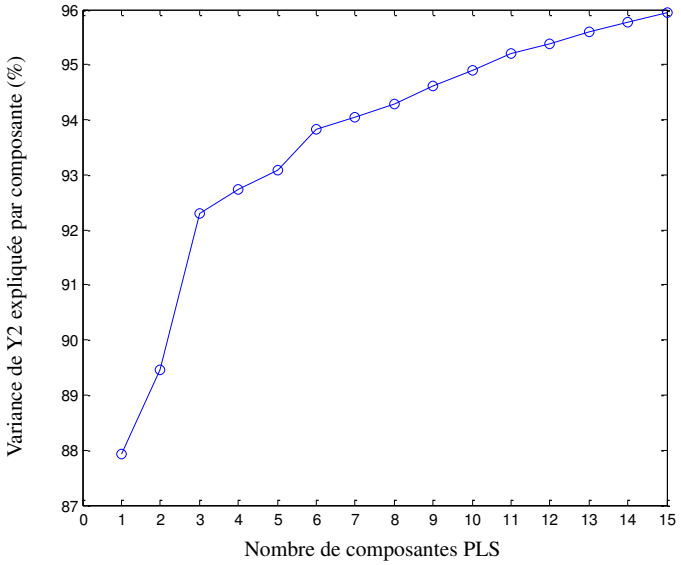


FIGURE 2.31 – Représentation du pourcentage de variance cumulée.

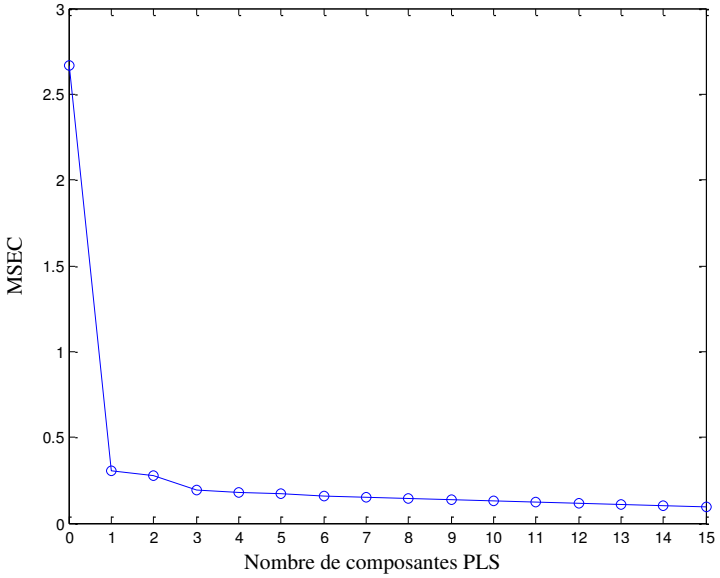


FIGURE 2.32 – Représentation du MSEC en fonction du nombre de composantes PLS.

A partir des figures 2.31 et 2.32 nous choisissons de ne retenir que 4 composantes PLS.

2.2.4.3 Calcul des critères *a posteriori*

Comme pour la réponse ”Y1”, nous proposons de calculer et de comparer les critères *a posteriori* des sous-ensembles de calibration (figure 2.33 et figure 2.35) et de validation (figure 2.34). Les valeurs de ces critères sont regroupées dans le tableau 2.10.

Tableau 2.10 – Tableau récapitulatif des critères *a posteriori* pour l'analyse de la réponse ”Y2”.

CALIBRATION (180 points)	VALIDATION (45 points)	RMSEC	R²cal	MAXcal	RMSEP	MAXval
Jeu n°1		0.424	0.933	1.56	0.405	1.08
Jeu n°2		0.4202	0.931	1.58	0.405	1.11
Jeu n°3		0.4202	0.931	1.58	0.42	1.11
Jeu n°4		0.3997	0.923	1.46	0.482	1.79
Jeu n°5		0.4272	0.910	1.68	0.378	0.91
Jeu n°6		0.4173	0.913	1.59	0.423	0.99
Jeu n°7		0.4195	0.92	1.60	0.410	0.94
Jeu n°8		0.4023	0.934	1.46	0.479	1.18
Jeu n°9		0.423	0.926	1.71	0.393	1.03
Jeu n°10		0.3857	0.94	1.19	0.529	1.77

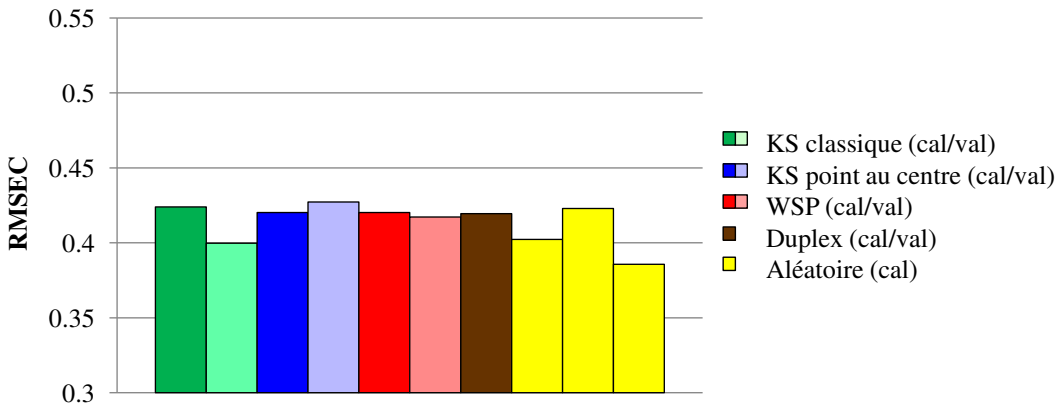


FIGURE 2.33 – RMSEC obtenus à partir des sous-ensembles de calibration pour l'analyse de la réponse ”Y2”.

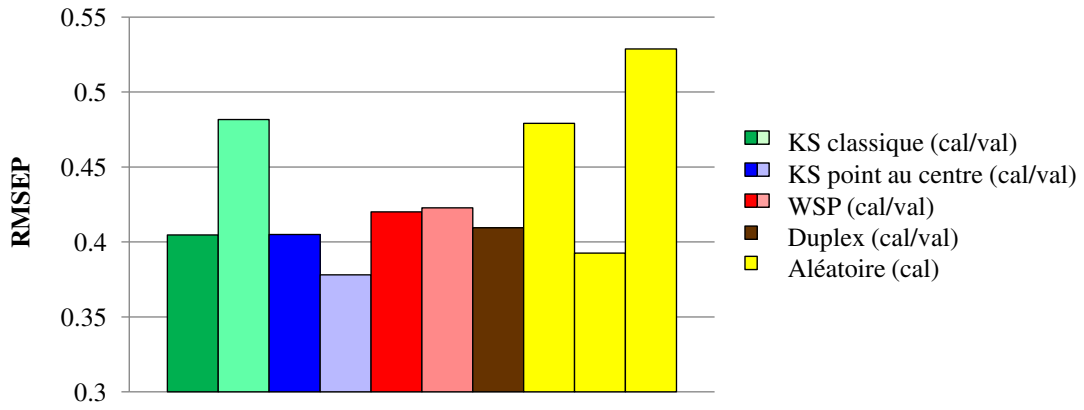


FIGURE 2.34 – RMSEP obtenus à partir des sous-ensembles de validation pour l'analyse de la réponse "Y2".

La figure 2.33 montre que toutes les méthodes de construction des sous-ensembles conduisent à un RMSEC quasiment identique, alors que les valeurs du RMSEP obtenues par les algorithmes de sélection se différencient un peu plus (figure 2.34).

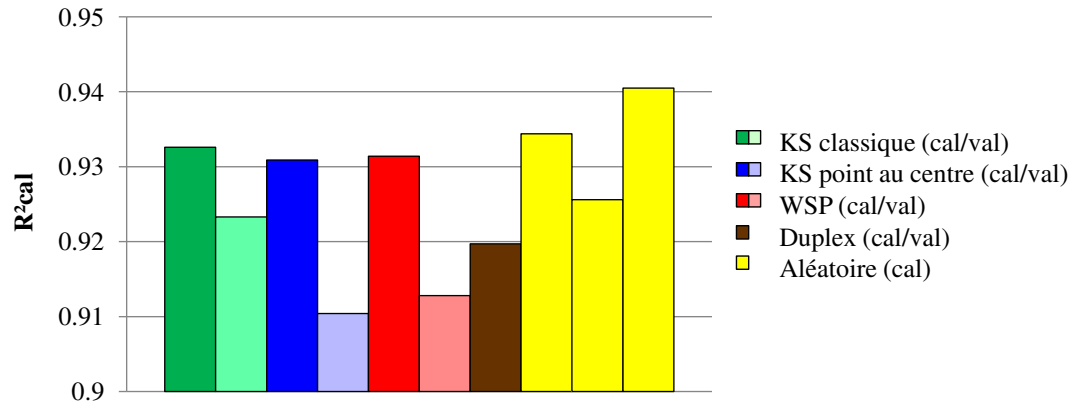


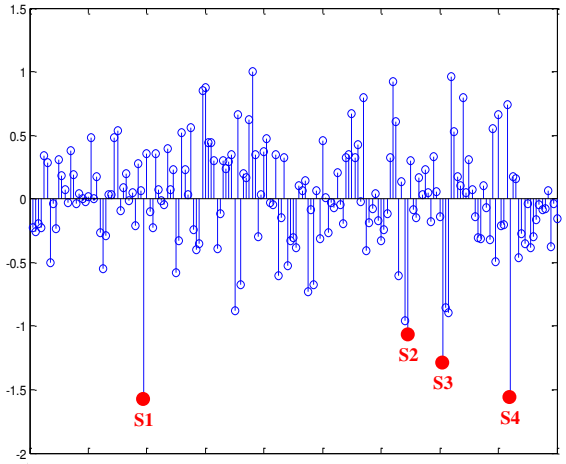
FIGURE 2.35 – R^2 calculés à partir des sous-ensembles de calibration pour l'analyse de la réponse "Y2".

Lors de l'analyse du coefficient R^2 des sous-ensembles de calibration (figure 2.35), les valeurs obtenues sont proches et présentent peu de variation selon la stratégie utilisée pour la construction. L'analyse des critères RMSEC et RMSEP ne permet pas de conclure sur l'efficacité d'une méthode de construction par rapport à une autre.

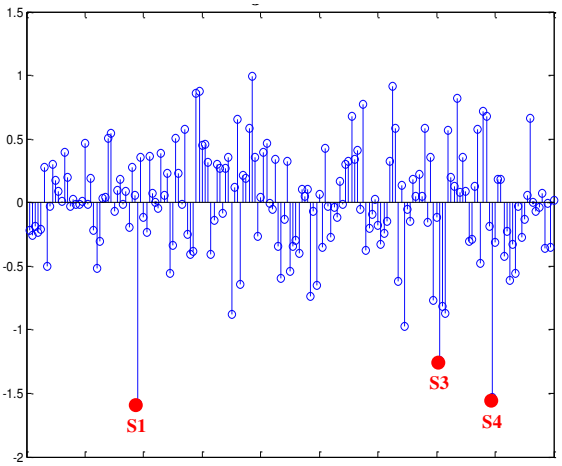
2.2.4.4 Analyse des résidus en Y

Les graphes ci-dessous permettent de visualiser les résidus en Y pour chaque stratégie de construction des sous-ensembles.

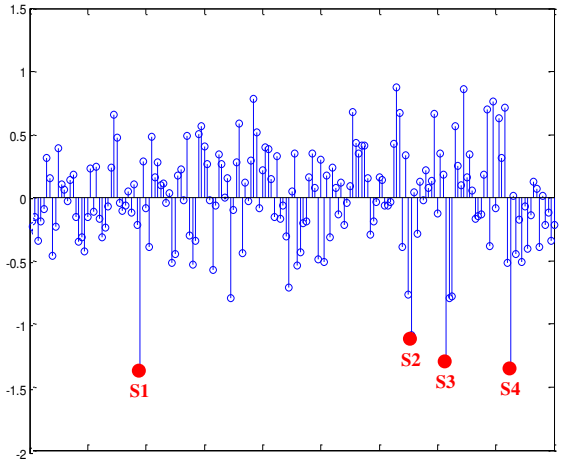
Jeu n°1 :



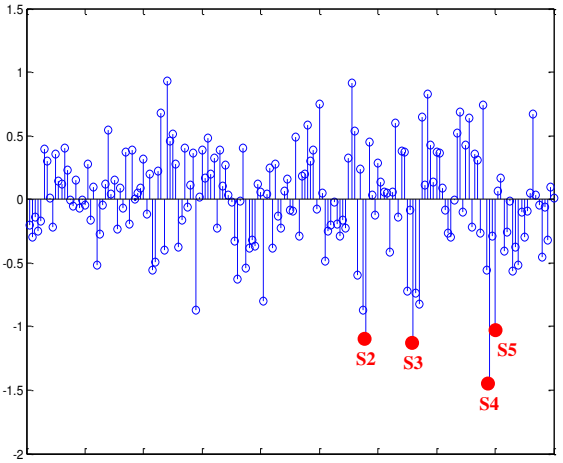
Jeu n°2 :



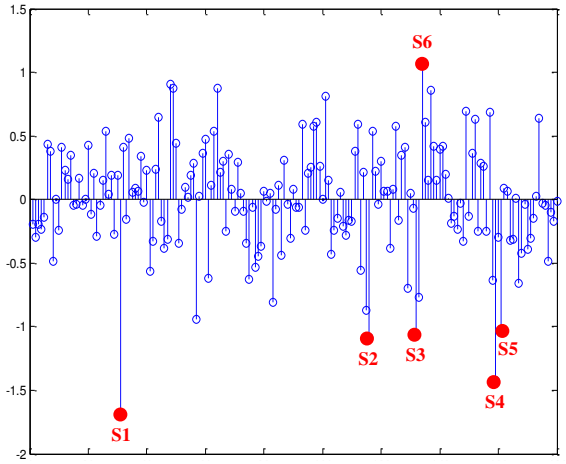
Jeu n°3 :



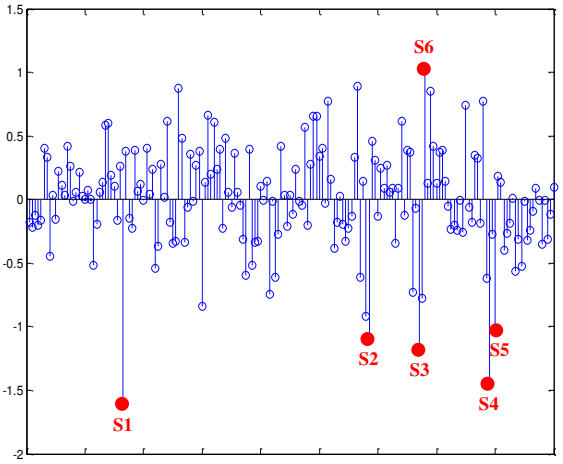
Jeu n°4 :



Jeu n°5 :



Jeu n°6 :



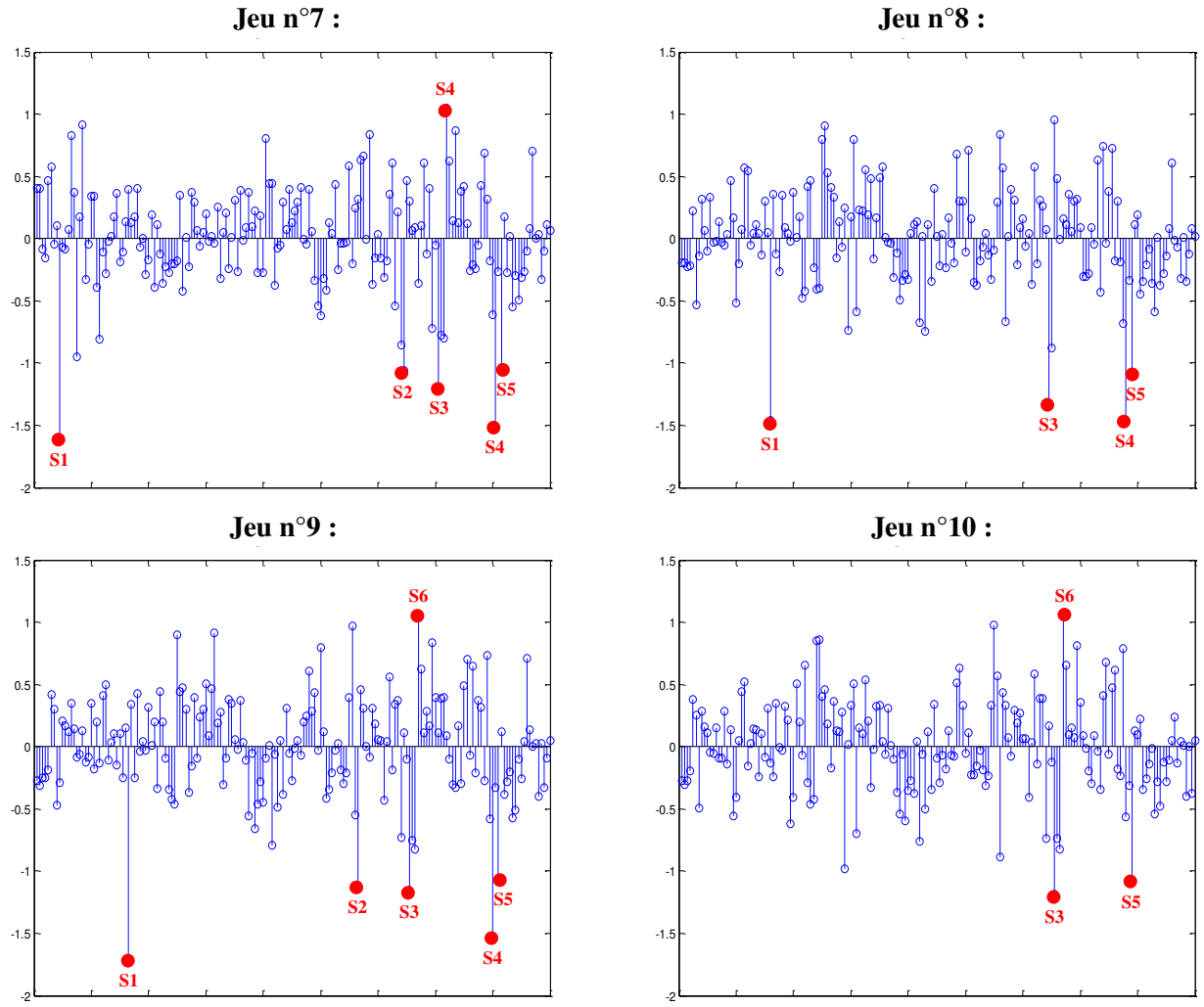


FIGURE 2.36 – Résidus en Y obtenus pour l'analyse de la réponse "Y2" par PLS.

La figure 2.36 montre que les résidus en Y les plus importants sont obtenus systématiquement pour les mêmes points. En effet, quelle que soit la méthode de construction envisagée pour construire les sets de calibration et de validation, nous retrouvons les spectres S1, S2, S3, S4, S5 et dans une moindre mesure le spectre S6. A ce stade de l'étude par simple observation des résidus nous avons considéré les 6 points cités ci-dessus comme des outliers en Y et nous avons choisi cette fois de reconstruire les sets de calibration à partir du nouvel ensemble de points candidats, constitué maintenant de $225 - 6 = 219$ spectres.

2.2.5 Étude de la réponse Y2 après suppression des outliers en Y

2.2.5.1 Construction et caractérisation des sous-ensembles de calibration et de validation

Les graphes ci-dessous représentent les valeurs des critères *a priori* des nouveaux sets de calibration (figure 2.37) et validation (figure 2.38).

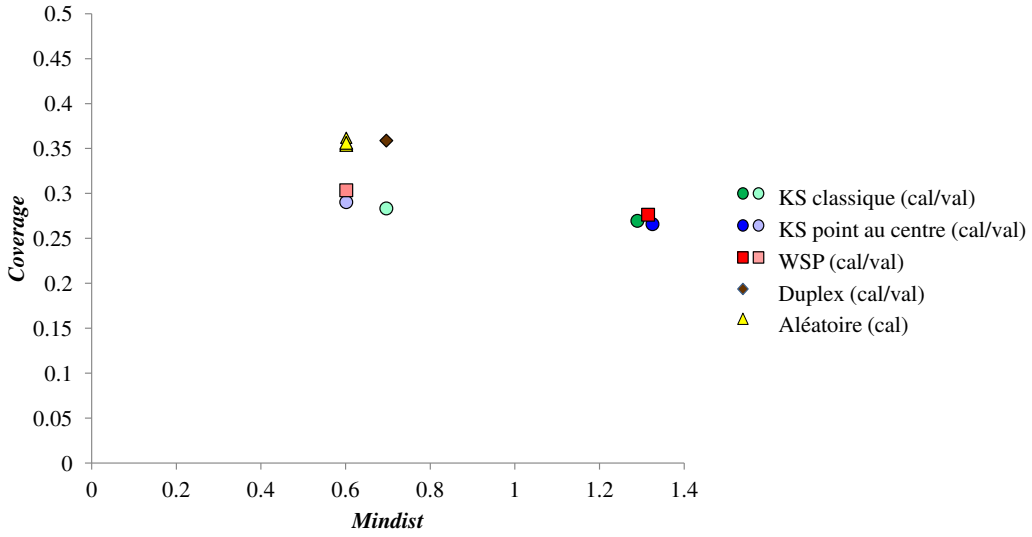


FIGURE 2.37 – Comparaison des critères intrinsèques de qualité des nouveaux **sous-ensembles de calibration** contenant 80% des données initiales ($N = 175$ points) sans les outliers en Y pour l'analyse de la réponse "Y2".

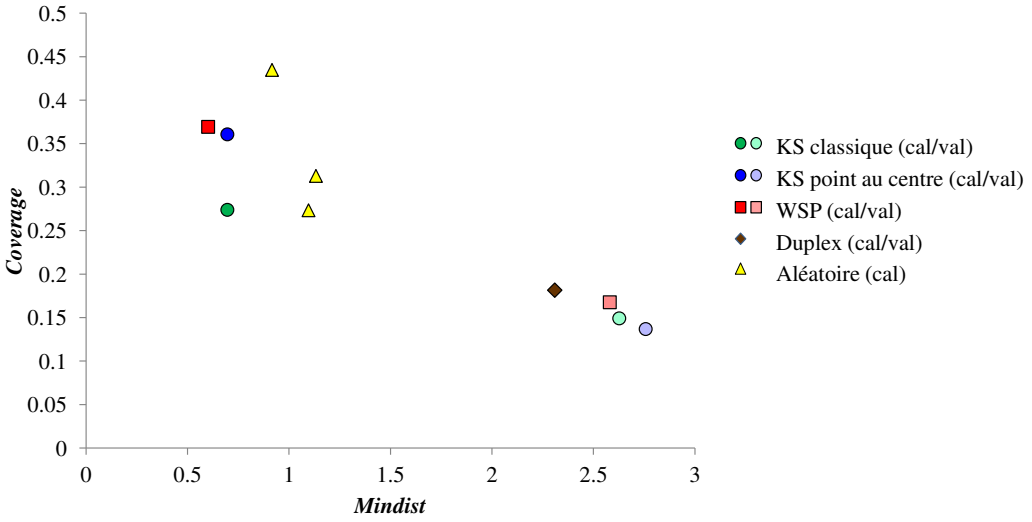


FIGURE 2.38 – Comparaison des critères intrinsèques de qualité des nouveaux **sous-ensembles de validation** contenant 20% des données initiales ($N = 44$ points) sans les outliers en Y pour l'analyse de la réponse "Y2".

Les valeurs des critères *Mindist* et *Coverage* pour tous les ensembles sont comparables aux valeurs obtenues avant la suppression des outliers, ce qui laisse supposer que leur positionnement dans l'espace n'était pas isolé.

2.2.5.2 Calculs des critères *a posteriori* à partir des modèles de régression PLS

Après l'étude des critères d'uniformité des différents sous-ensembles, nous proposons d'étudier les critères *a posteriori* des modèles de régression PLS. Les valeurs de ces critères sont rappelées dans le tableau 2.11.

Tableau 2.11 – Tableau récapitulatif des critères *a posteriori* calculés sur les 219 spectres candidats pour l'analyse de la réponse "Y2" après suppression des outliers en Y.

CALIBRATION (175 points)	VALIDATION (44 points)	RMSEC	R²cal	MAXcal	RMSEP	MAXval
Jeu n°1		0.36	0.951	0.951	0.387	0.819
Jeu n°2		0.37	0.946	0.971	0.329	0.8
Jeu n°3		0.39	0.948	0.974	0.359	0.833
Jeu n°4		0.36	0.934	0.913	0.374	0.984
Jeu n°5		0.36	0.935	0.984	0.360	0.935
Jeu n°6		0.36	0.937	0.929	0.399	1.016
Jeu n°7		0.35	0.356	0.994	0.396	0.962
Jeu n°8		0.37	0.944	1.005	0.343	0.927
Jeu n°9		0.36	0.949	1.05	0.371	0.849
Jeu n°10		0.37	0.941	1.000	0.345	0.825

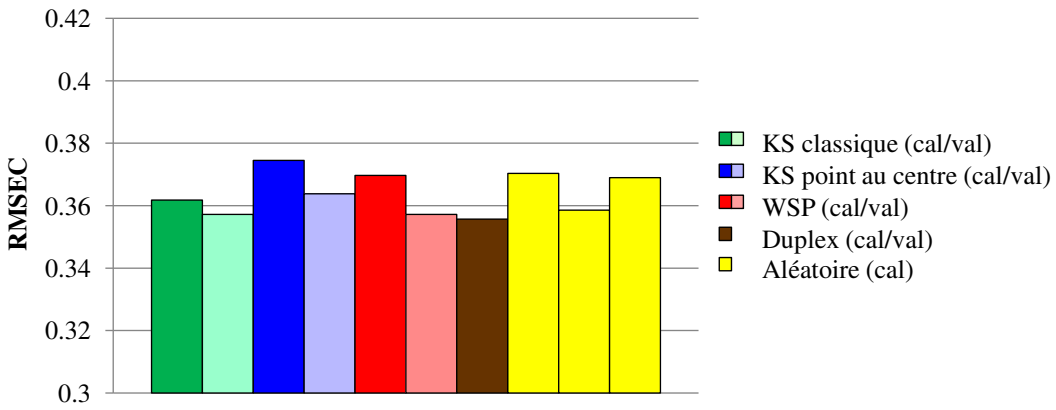


FIGURE 2.39 – RMSEC obtenus à partir des nouveaux sous-ensembles de calibration pour l'analyse de la réponse "Y2".

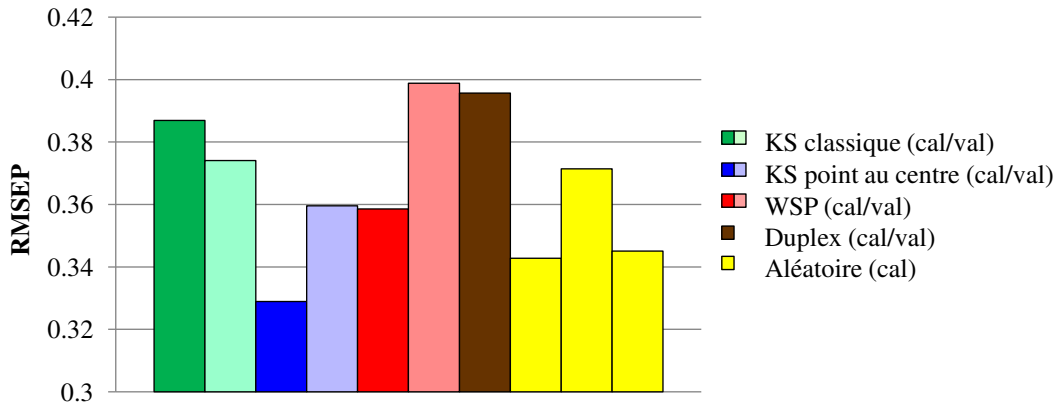


FIGURE 2.40 – RMSEP obtenus à partir des nouveaux sous-ensembles de validation pour l'analyse de la réponse "Y2".

Les graphes ci-dessus montrent une légère amélioration des performances du modèle. En effet, on observe une diminution des critères RMSEC et RMSEP à partir des nouveaux sets et ce, indépendamment de la stratégie de construction.

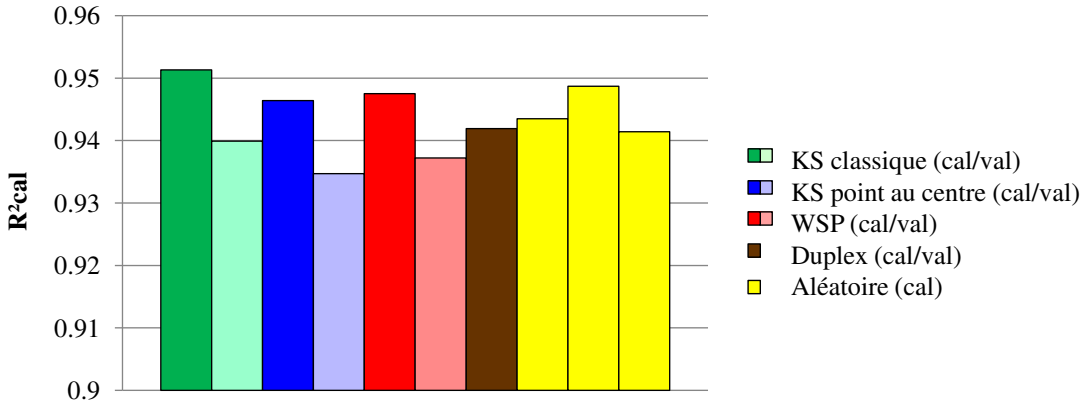


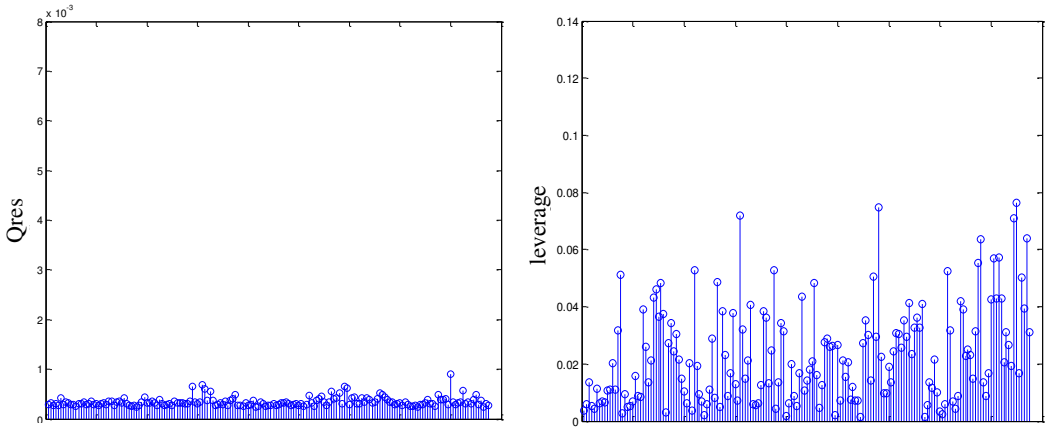
FIGURE 2.41 – R^2 obtenus à partir des nouveaux sous-ensembles de calibration pour l'analyse de la réponse "Y2".

De même, la comparaison des coefficients R^2 des sets de calibration (figure 2.41) met en évidence des valeurs qui sont plus élevées et quelle que soit la stratégie envisagée ces valeurs sont proches de 0.94.

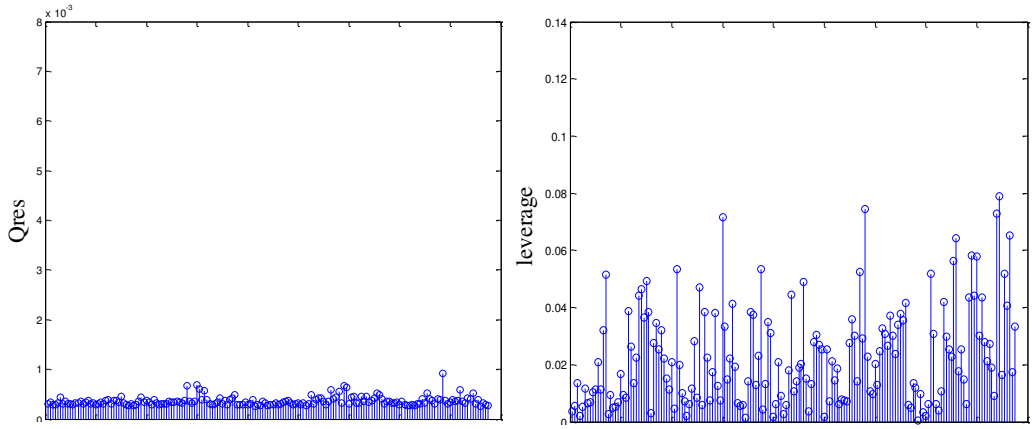
2.2.5.3 Analyse des résidus en X après suppression des outliers en Y

La première partie de l'étude de la réponse "Y2" nous a permis de détecter les outliers en Y. La construction des modèles de régression PLS sans prendre en compte ces outliers en Y conduit à une légère amélioration des critères *a posteriori*. Nous souhaitons compléter cette étude par l'analyse des résidus en X afin de détecter les outliers spectraux (figure 2.42) à partir des nouveaux modèles.

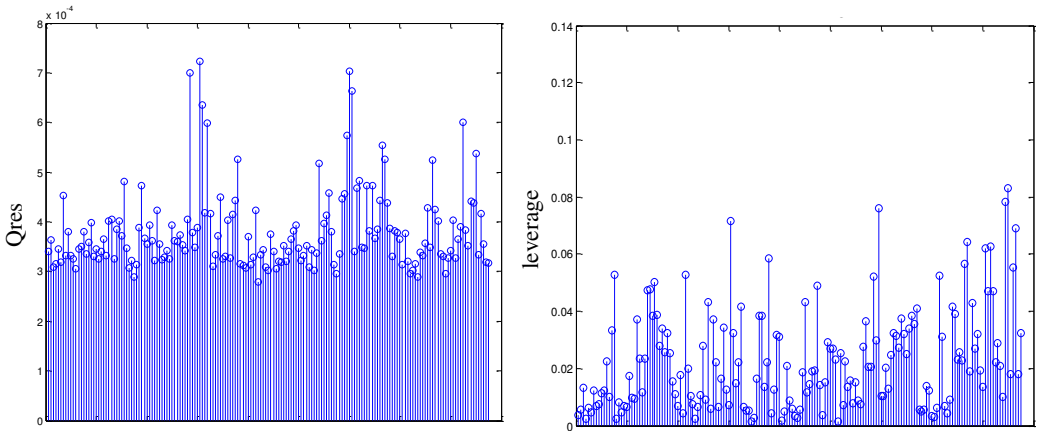
Jeu n°1 :



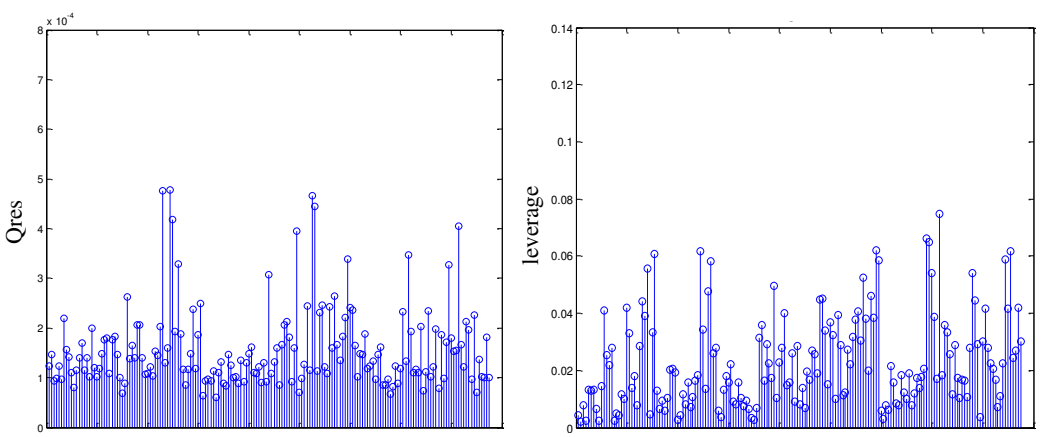
Jeu n°2 :



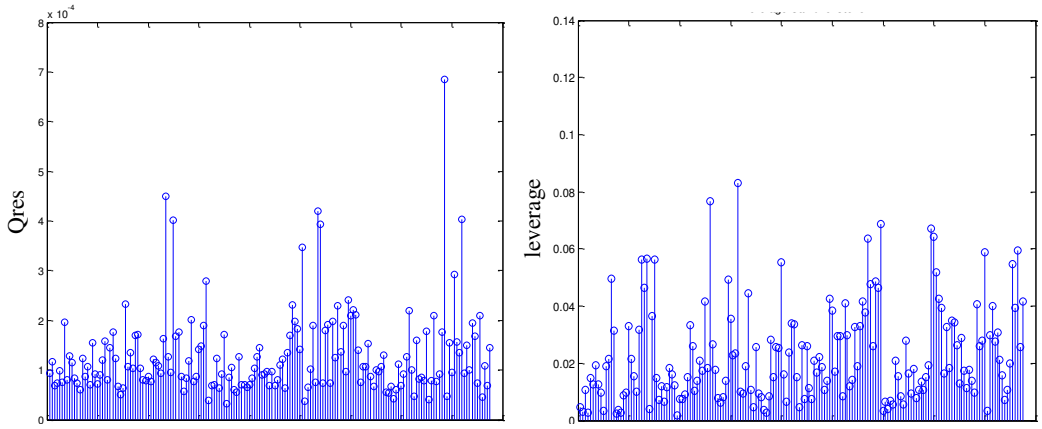
Jeu n°3 :



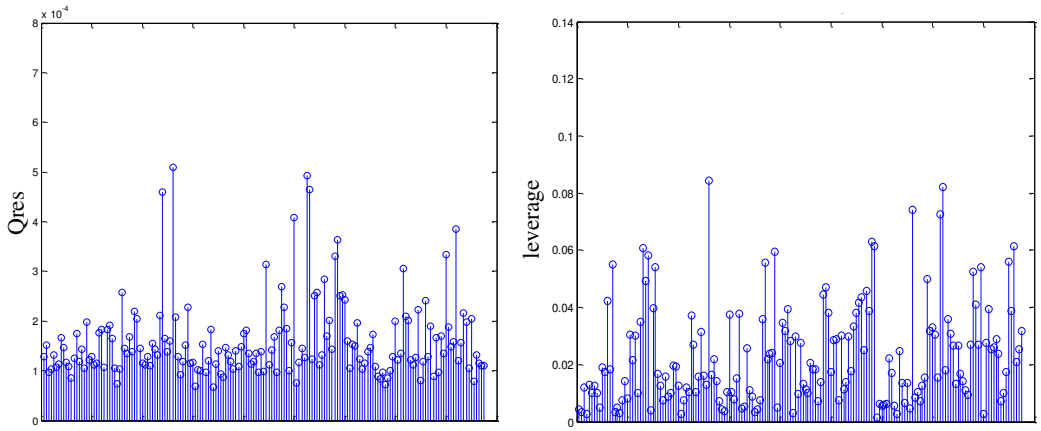
Jeu n°4 :



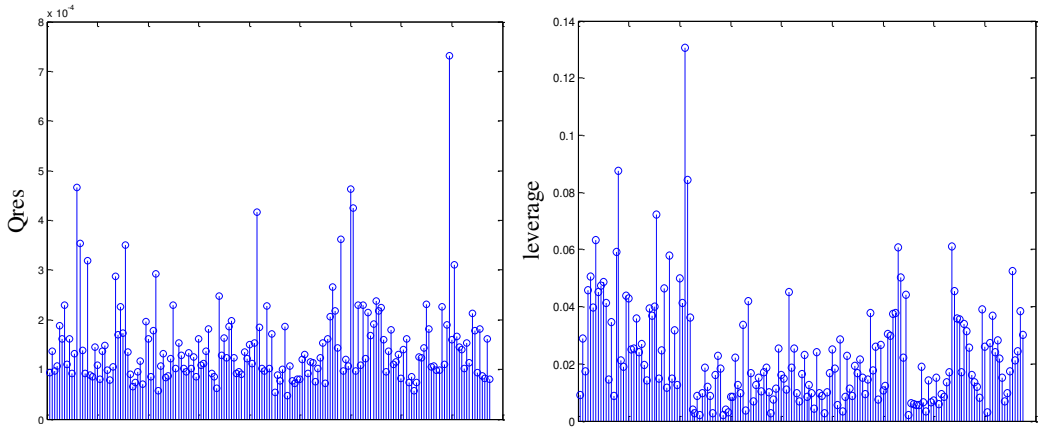
Jeu n°5 :



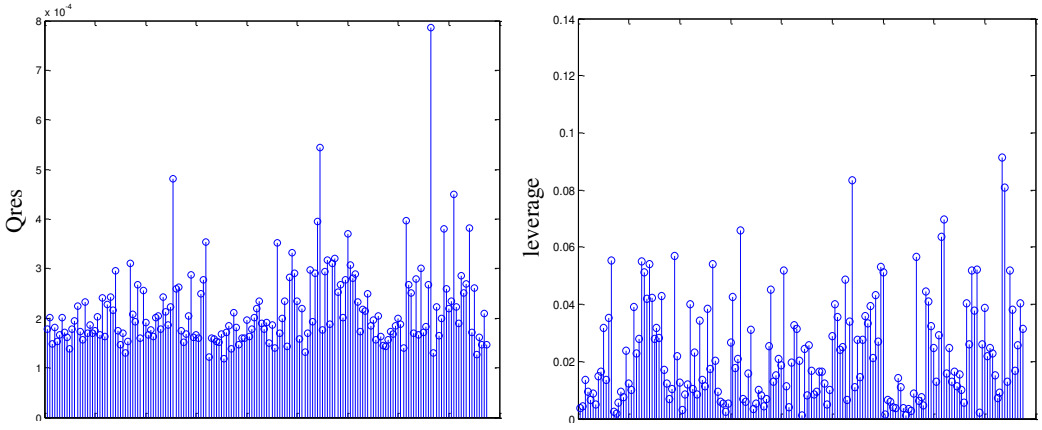
Jeu n°6 :



Jeu n°7 :



Jeu n°8 :



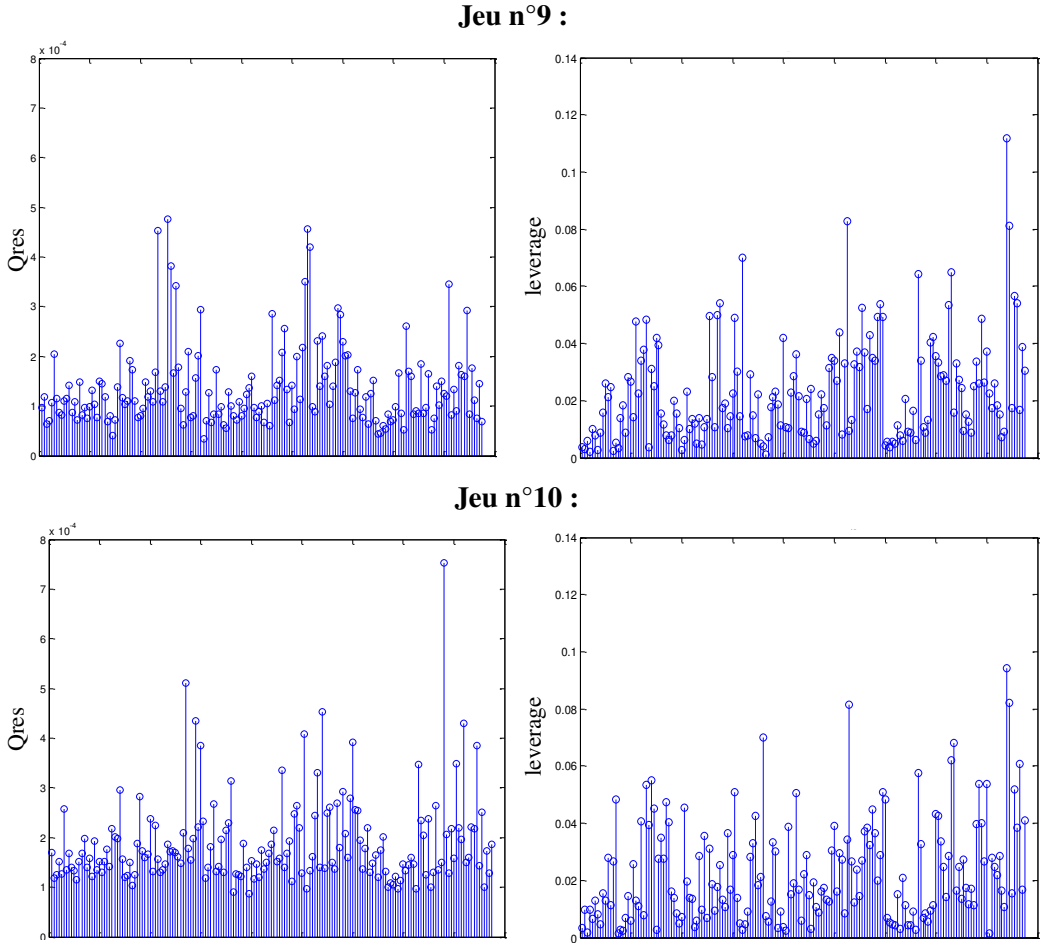


FIGURE 2.42 – Résidus en X obtenus à partir des nouveaux sous-ensembles pour l'analyse de la réponse "Y2" par PLS.

Sur la figure 2.42, nous n'observons pas d'outliers évidents en X. En effet, les faibles valeurs des critères *Qres* et *leverage* conduisent à peu de variation de ces critères entre les différents spectres.

2.2.6 Conclusion

Dans le cadre de cette étude, nous avons comparé différentes stratégies de construction des sous-ensembles de calibration et de validation pour des modèles de régression PLS. Les différentes méthodes de sélection de points (en calibration ou en validation) reposent sur les algorithmes de Kennard et Stone (KS classique ou départ au centre du domaine), WSP, Duplex et une sélection aléatoire. La comparaison des performances des modèles calculés à partir des différents sous-ensembles de points (sélectionnés par les différentes méthodes citées) est réalisée à partir des valeurs de RMSE des modèles de régression PLS. Cette comparaison ne montre pas de véritable différence entre les méthodes, si ce n'est qu'il est préférable de construire un set de calibration bien conditionné en termes d'uniformité de remplissage de l'espace pour avoir une prévision de bonne qualité (RMSEC ~ RMSEP). D'autre part, une simple sélection aléatoire des points ne conduit pas systématiquement à des modèles moins performants, mais demeure sensible au tirage initial et conduit donc à des résultats plus ou moins bons. Pour ce qui est des différents algorithmes de sélection, les algorithmes tels que KS, WSP et Duplex, de par leur principe de construction, vont permettre de sélectionner pour le set de calibration des points

extrêmes dans l'espace des variables, susceptibles de présenter des comportements particuliers qu'il est important de considérer dans l'étape d'apprentissage. Ces algorithmes conduisent à des résultats similaires à la fois pour les performances du modèle et pour la détection des outliers. Cependant, nous rappelons que l'algorithme KS est plus long en temps de calcul que l'algorithme WSP et cette différence de temps est accentuée lorsque la dimension et le nombre de points sélectionnés augmentent. A ce stade de l'étude, il semble difficile de relier les critères *a priori*, caractérisant les sous-ensembles sélectionnés par algorithme en termes d'uniformité, aux critères *a posteriori* quantifiant la qualité du modèle. Nous avons également étudié l'impact de la présence d'outliers en X et/ou en Y dans le modèle. Pour cela, une première régression PLS a été effectuée en considérant tous les points des sets de calibration puis nous avons effectué une nouvelle régression après avoir supprimé les spectres détectés comme outliers.

A partir de l'étude des réponses "Y1" et "Y2", nous n'avons pas pu mettre en évidence une nette amélioration de la qualité du modèle lorsque les outliers sont retirés du set de calibration, ce qui peut probablement s'expliquer par un "appauvrissement" de ce sous-ensemble de points au sens de la représentativité de l'ensemble des points initiaux. Aussi, nous recommandons de reconstruire de nouveaux sets de calibration/validation après la suppression des outliers des points candidats, pour garantir un bon conditionnement de ces sous-ensembles.

2.3 Conclusion de l'analyse des données spectroscopiques

A partir de ces deux exemples issus de la spectroscopie infrarouge, nous avons proposé d'utiliser les méthodes appartenant au "catalogue" afin de travailler avec ces données en grande dimension.

Dans la première étude, nous souhaitions identifier les spectres IR qui appartenaient au même groupe. Pour cela, nous avons utilisé l'ACC qui, avec une simple représentation graphique en 2D, nous a permis d'identifier 4 groupes sur 5, avec un groupe réunissant deux familles de fromages. Cependant, il semble difficile de séparer ce dernier groupe puisqu'il se compose de deux familles avec des compositions chimiques analogues ce qui peut expliquer pourquoi par l'ACC, nous ne pouvons pas les différencier.

La deuxième application avait pour objectif de comparer les méthodes de sélection de points, notamment les algorithmes de Kennard et Stone (classique ou avec départ du point au centre du domaine), WSP, Duplex et une simple sélection aléatoire pour la construction des sous-ensembles de calibration et de validation en vue d'établir un modèle de régression PLS. Nous avons alors proposé deux approches afin de vérifier l'importance de la présence d'outliers en Y et/ou en X dans les sets destinés à l'apprentissage du modèle. La première approche consistait à construire les sous-ensembles puis de proposer un modèle de régression PLS, identifier les outliers en Y, supprimer du sous-ensemble de calibration les outliers en Y, reconstruire un modèle de régression PLS à partir du nouveau set de calibration. La deuxième se différencie par la reconstruction des sets de calibration et de validation à partir de la base de données à laquelle nous avons supprimé les outliers en Y. Les résultats de ces deux approches ne permettent pas de proposer une méthode universelle pour la construction des sous-ensembles et ainsi obtenir le modèle de meilleure qualité mais nous avons montré qu'il est préférable de les reconstruire si nous supprimons les spectres détectés comme outliers. En effet, la suppression d'un outlier d'un ensemble de calibration crée probablement des zones déficientes en information dans l'espace ce qui modifie alors le bon conditionnement du sous-ensemble destiné à l'apprentissage.

Chapitre 3

Applications à la simulation numérique

Dans de nombreux domaines tels que la pétrochimie, l'astronomie, la météorologie, ... il est d'usage d'utiliser des modèles de simulation pour représenter au mieux des phénomènes réels via des codes de calcul. Au travers d'une approche plus fine des phénomènes physiques étudiés, les codes de calcul deviennent de plus en plus réalistes en considérant un grand nombre de variables d'entrée et peuvent être onéreux en temps de calcul.

Il est alors nécessaire d'élaborer une stratégie optimale permettant d'obtenir des informations indispensables comme un classement par ordre d'importance des variables d'entrée du modèle ou une idée sur l'allure générale de la surface de réponse à approcher. Cette stratégie doit être la plus efficace possible et doit garantir une information de bonne qualité, même en grande dimension. L'utilisation de plans d'expériences pour choisir au mieux les simulations numériques à réaliser semble être adaptée à cette problématique et leur usage est de nos jours de plus en plus fréquent. Mais le nombre de variables d'entrée, souvent très grand (plusieurs dizaines, voire centaines) et les larges domaines de variation, font que les plans d'expériences classiques ne sont plus vraiment appropriés car les critères mathématiques associés à ces plans imposent une répartition des points (des simulations) aux extrémités du domaine de variation des facteurs. En simulation numérique, on préfère utiliser des plans d'expériences appelés Space-Filling Designs (SFD) [80, 81, 82] ou plans uniformes qui vont répartir les points uniformément dans l'espace des variables d'entrée. Toutefois, tous les SFD ne sont pas équivalents en termes de critères de qualité qui mesurent l'uniformité de la répartition des points, comme les valeurs des critères intrinsèques définis précédemment, à savoir les valeurs *Mindist* [5, 6, 7], *Coverage* [8]. De plus de nombreux algorithmes, performants en faible dimension ($D < 10$), se révèlent beaucoup moins efficaces à grande dimension ($D > 20, 30, ...$). Ainsi, les suites à faible discrépance [83, 84, 85, 86, 87] comme les suites de Faure présentent, en grande dimension, de très mauvais critères d'uniformité avec des valeurs *Mindist* faibles et *Coverage* élevées. Le mauvais conditionnement de ces plans d'expériences se caractérise par une répartition non uniforme des points dans l'espace, synonyme d'une accumulation ou d'une déficience de points en certaines zones de l'espace, que nous qualifierons respectivement "d'amas" et de "lacunes". Comme nous le verrons plus tard, un mauvais conditionnement au sens d'une répartition non uniforme peut aussi se rencontrer dans une autre situation et plus précisément dans le cas où l'on examine l'uniformité de la répartition de points dans un espace de dimension réduite après une analyse de sensibilité. En effet, une analyse de sensibilité vise à identifier les variables d'entrée qui contribuent fortement à la variabilité de la réponse [88]. Il peut être judicieux alors de simplifier le modèle en ne considérant que les variables influentes et ainsi de réduire la dimension de l'espace pour réaliser une étude

plus fine du phénomène (modélisation) mais avec comme objectif de conserver les essais (simulations) déjà réalisés. Cette réduction de l'espace appelée "repliage" du plan, qui consiste à projeter les points d'un espace de dimension D dans un espace de dimension réduite D', ne présente aucune garantie sur l'uniformité de la répartition des points ; en effet, elle peut engendrer des amas de points ou des lacunes dans le nouvel espace d'intérêt.

L'objectif de ce travail porte sur la réparation de plans dont les points ne sont pas répartis uniformément dans le domaine des variables soit du fait d'une mauvaise construction, soit après un "repliage" du plan initial. Nous présenterons tout d'abord les difficultés qui peuvent être rencontrées lors de la construction de plans uniformes en grande dimension, puis nous appliquerons les algorithmes de réparation sur un exemple en deux dimensions pour suivre visuellement les différentes étapes, pour ensuite nous intéresser aux SFD en vingt dimensions. Enfin, un dernier exemple illustrera le "repliage" en étudiant les conséquences sur la qualité intrinsèque des plans projetés en 10, 30 et 50 dimensions pour lesquels nous ferons varier la taille du sous-ensemble de facteurs conservés.

3.1 État de l'art des plans uniformes

Il existe plusieurs familles de plans uniformes, qui se différencient dans leur construction par le choix du critère à optimiser, parmi lesquels nous pouvons citer :

- les suites à faible discrépance, qui utilisent un algorithme déterministe pour obtenir une distribution uniforme des points, basé sur le critère de la discrépance [89] mesurant la distance entre une distribution de points empirique et une distribution de points théorique. Leur construction utilise une fonction radiale inverse en base b, b étant un entier positif et repose sur les suites de Van Der Corput [90, 89]. Une étude bibliographique a montré que ces suites ne sont pas optimales en termes d'uniformité dès que le nombre de dimensions augmente. Cependant, leur construction est simple et rapide même en grande dimension, ce qui permet de les envisager comme simple algorithme de génération de points. Parmi les suites à faible discrépance, nous citons seulement celles que seront utilisées dans la suite de ce chapitre, à savoir :
  - les suites de Faure [83], qui sont définies à partir d'une base b unique, avec b un nombre entier supérieur à D. Généralement la valeur de b correspond au nombre premier supérieur ou égal à D,
  - les suites de Halton [84], qui sont une généralisation des suites de Van Der Corput avec $D \geq 1$. L'idée principale est d'utiliser une base différente pour chaque dimension,
  - les suites de Hammersley [85], qui en D dimensions sont construites à partir d'un terme dépendant du nombre de points N et d'une suite de Halton de dimension $(D-1)$.
  - les suites de Sobol [86, 87, 91], qui requièrent l'utilisation de polynômes primitifs P_j de degré s_j le plus faible possible :

$$P_j = x^{s_j} + a_{1,j}x^{s_j-1} + a_{2,j}x^{s_j-2} + ... + a_{s_j-1,j}x + 1$$

    avec $a_{i,j}$, les coefficients du polynôme qui prennent des valeurs égales à 0 ou 1 et x, qui représente une suite. Ainsi, construire une suite de Sobol en D dimensions nécessite de choisir D polynômes primitifs distincts.

- les hypercubes latins [92, 93, 94, 95, 96], qui se caractérisent par une construction selon une contrainte de projection des points sur les axes des variables d'entrée. Par définition, chaque axe est divisé en N intervalles de même longueur de la façon suivante : $\left\{\left[0, \frac{1}{N}\right], \left[\frac{1}{N}, \frac{2}{N}\right], ..., \left[\frac{N-1}{N}, 1\right]\right\}$ conduisant ainsi à un maillage de taille $N \times D$. Chaque colonne de l'hypercube latin, à N points et de dimension D, est une permutation aléatoire de $\{1, ..., N\}$. Les points d'un hypercube latin ont la propriété intéressante d'être uniformément distribués sur les axes factoriels mais cette propriété n'assure pas toujours un bon remplissage de l'espace. Pour pallier ces problèmes, la construction de ces plans peut être optimisée selon différents critères : par exemple, l'amélioration du critère de corrélation conduit à des hypercubes latins orthogonaux [97, 94, 98] ou encore à des hypercubes latins maximin lorsque la distance minimale entre les points est maximisée [5],

- les plans de Strauss [99, 100], qui sont générés selon le processus ponctuel de Gibbs [101] qui repose sur le phénomène de répulsion entre particules. Dans la construction des

SFD, la transposition de ce processus revient à considérer dans un espace, N particules de même charge à un instant t comme un ensemble de N points. Ce processus de répulsion fait intervenir deux paramètres : le rayon R de la particule et la probabilité d'interaction entre particules γ. Dans le cas des SFD, le rayon de la particule est assimilé au rayon d'une hypersphère et l'interaction à un chevauchement d'hypersphère. C'est à travers ce paramètre que réside la répulsion. Plus précisément, ce paramètre est une contrainte sur la dimension 1 et sera noté γ_{1D}. Le choix de ces paramètres joue un rôle primordial car ils conditionnent la distribution des points dans l'espace. En effet, un rayon R trop petit crée des lacunes et à l'inverse, une valeur trop grande peut conduire à des amas. Une valeur γ_{1D} faible favorise une répartition uniforme dans l'espace, mais une valeur proche de zéro conduit à des alignements de points car les hypersphères sont qualifiées de "dures". Pour pallier cette contrainte, un nouveau paramètre a été ajouté : l'interaction d'ordre général, notée γ. La construction de tels plans se fait selon des itérations par la méthode Monte Carlo par chaînes de Markov. Pour construire de tels plans, il est nécessaire de trouver le meilleur compromis entre les quatre paramètres : le rayon de l'hypersphère (R), les deux contraintes de répulsion (γ_{1D} et γ) et le nombre d'itérations, qui ont une influence directe sur les critères intrinsèques du plan et donc sur sa qualité,

– les plans de Audze-Eglais [3], qui optimisent un critère de répulsion, noté AE, en assimilant les points du plan à des particules chargées. Les points sont alors distribués selon la magnitude des forces de répulsion qui est inversement proportionnelle au carré de la distance euclidienne entre les points :

$$AE = \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \frac{1}{dist(x_i, x_k)^2}$$

avec $dist(x_i, x_k)$ la distance euclidienne entre deux points quelconques x_i et x_k

– les plans minimax et maximin [5], qui respectivement cherchent à minimiser les distances maximales ou à maximiser les distances minimales entre points,

– les plans construits par l'algorithme de sélection WSP [33, 34, 35, 36], qui permet d'extraire d'un ensemble de points, un sous-ensemble dont la répartition est la plus uniforme possible dans l'espace des variables, les points sélectionnés étant au moins séparés d'une distance d_{min} prédéfinie.

La majorité de ces algorithmes permet d'obtenir des plans uniformes de bonne qualité en faible dimension ($D < 10$) mais ils deviennent difficiles à construire ou se révèlent moins performants dès que la dimension devient grande ($D > 20$ ou 30). Par exemple, les suites à faible discrépance présentent de mauvais critères d'uniformité en grande dimension, avec de faibles valeurs *Mindist* et des valeurs *Coverage* élevées. Le mauvais conditionnement de ces plans se caractérise par une distribution non homogène des points dans l'espace des variables, qui se manifeste par des alignements de points et/ou des lacunes.

A titre comparatif, nous pouvons calculer les critères intrinsèques de quelques SFD classiques tels que les suites à faible discrépance, un hypercube latin aléatoire (rLHS), un plan de Strauss, un plan WSP, auxquels nous ajoutons un plan aléatoire. Chaque plan est construit en 2, 10, 20, 30, 40 et 50 dimensions en fixant 20 points par dimension. Nous utiliserons alors les représentations graphiques $Coverage = f(Mindist)$ et $Coverage = f(MoyMin)$ qui sont regroupées dans les figures 2.43 à 2.49 pour comparer les qualités intrinsèques de ces plans.

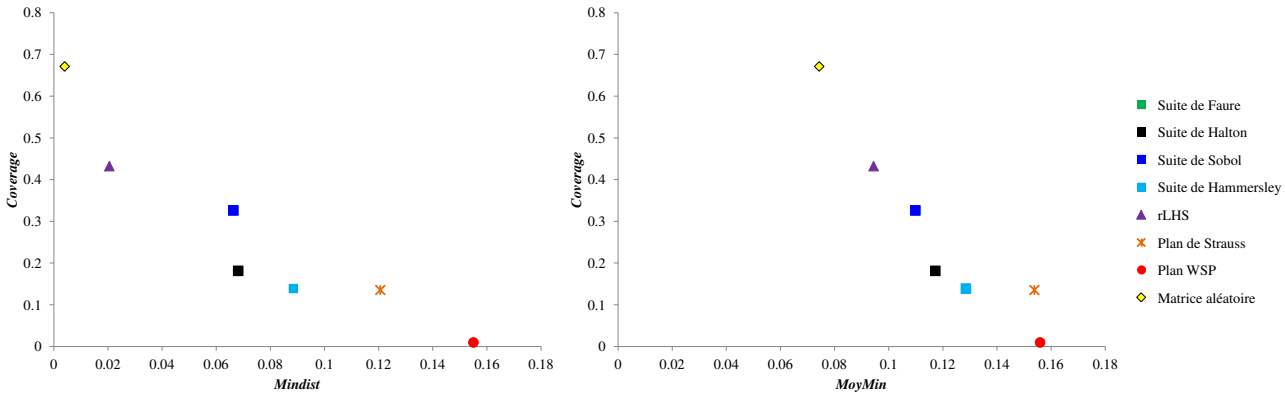
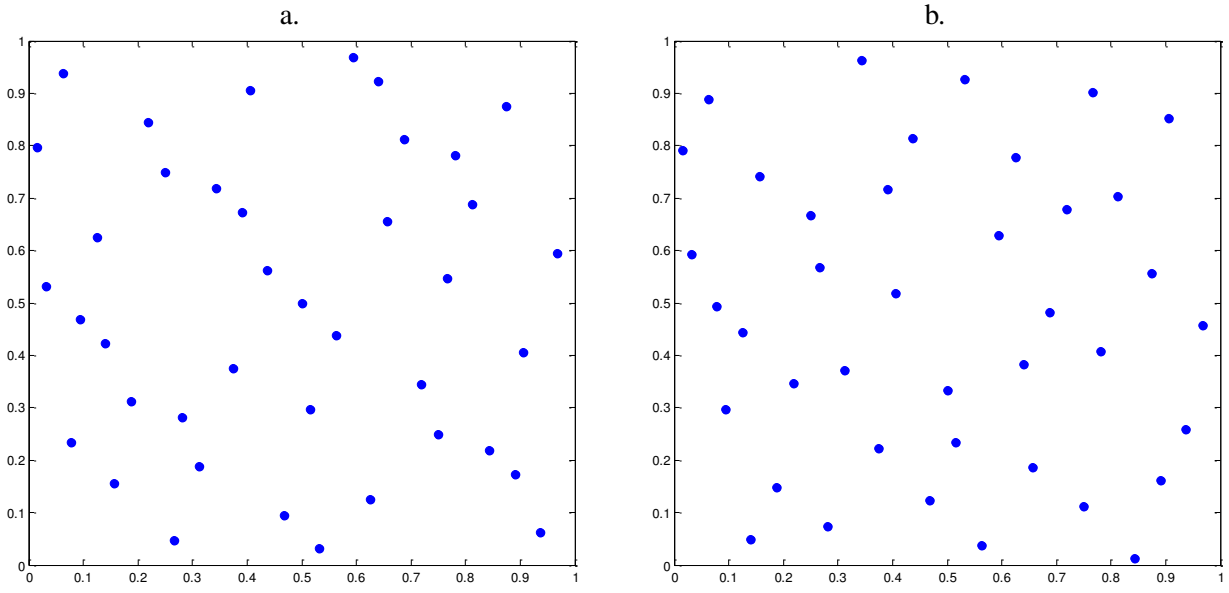


FIGURE 2.43 – Comparaison de la qualité intrinsèque de quelques SFD construits en 2D avec 40 points. Dans cette représentation graphique, la suite de Faure et la suite de Sobol ne peuvent pas être différenciées car elles présentent les mêmes valeurs de critères.

Nous observons une grande disparité dans les valeurs des critères en fonction de la nature du plan. Pour comprendre cette variabilité, nous pouvons représenter la répartition des points pour les différents SFD en 2D avec 40 points (figure 2.43).



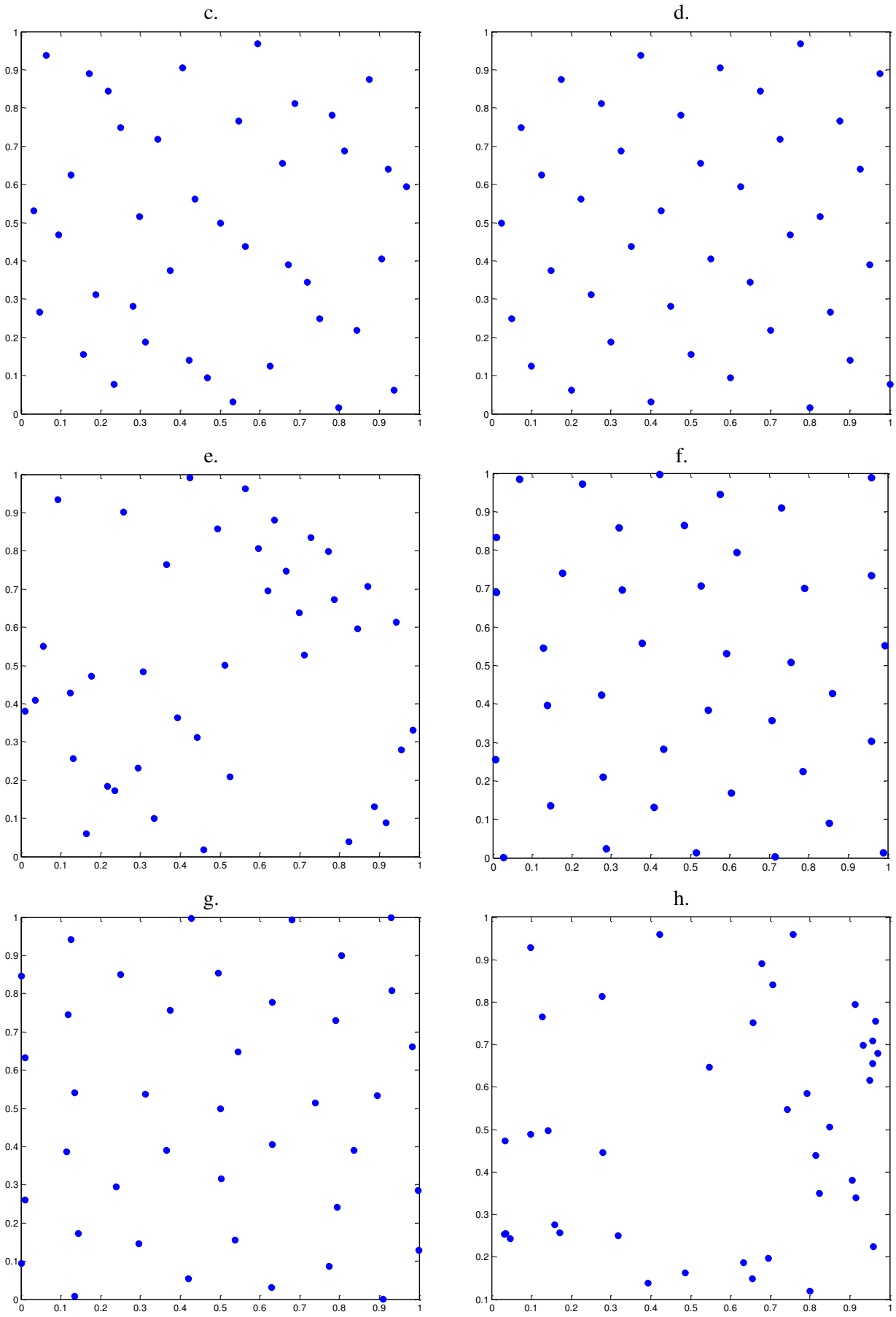


FIGURE 2.44 – Comparaison de la répartition des points des SFD en 2D avec 40 points. **a)** Suite de Faure, **b)** Suite de Halton, **c)** Suite de Sobol, **d)** Suite de Hammersley, **e)** Plan rLHS, **f)** Plan de Strauss, **g)** Plan WSP, **h)** Matrice aléatoire.

Nous observons que les plus mauvais plans sont le plan aléatoire et le plans rLHS qui présentent les plus faibles valeurs *Mindist*, inférieures aux valeurs *MoyMin*, et des valeurs *Coverage* élevées. Les représentations graphiques montrent que ces plans présentent des amas de points. Les suites à faible discrépance constituent un groupe caractérisé par des valeurs *Mindist* et *MoyMin* grandes et de faibles valeurs *Coverage*. Ces plans, de qualité intermédiaire, présentent des lacunes ou des rapprochements de points sans constituer pour autant des amas. Les plans de Strauss et WSP, qui présentent les meilleurs critères intrinsèques avec des valeurs *Mindist* et *MoyMin* élevées, montre un espace uniformément rempli par les points c'est-à-dire sans amas et sans lacunes mais la valeur *Mindist* du plan de Strauss est plus faible ce qui caractérise des rapprochements de points.

Ainsi, les informations qui sont apportées par le calcul des critères intrinsèques des SFD et les représentations graphiques sont en accord quant à la répartition des points dans l'espace des variables. En deux dimensions, nous pouvons représenter ces SFD mais lorsque la dimension sera plus grande, nous devrons nous limiter à la seule interprétation des critères intrinsèques. Les figures 2.45 à 2.49 permettent de comparer les valeurs des critères intrinsèques pour les SFD en 10D, 20D, 30D, 40D et 50D.

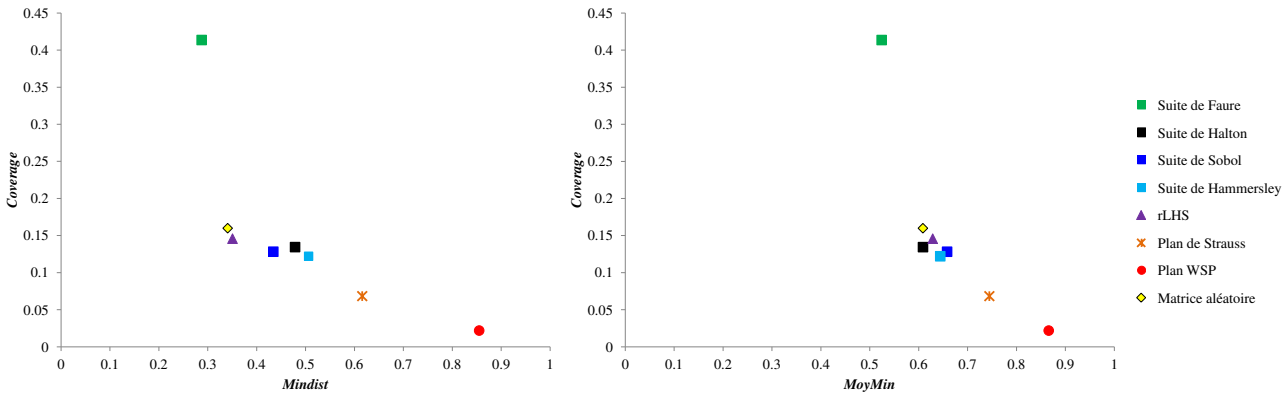


FIGURE 2.45 – Comparaison de la qualité intrinsèque de quelques SFD construits en 10D avec 200 points.

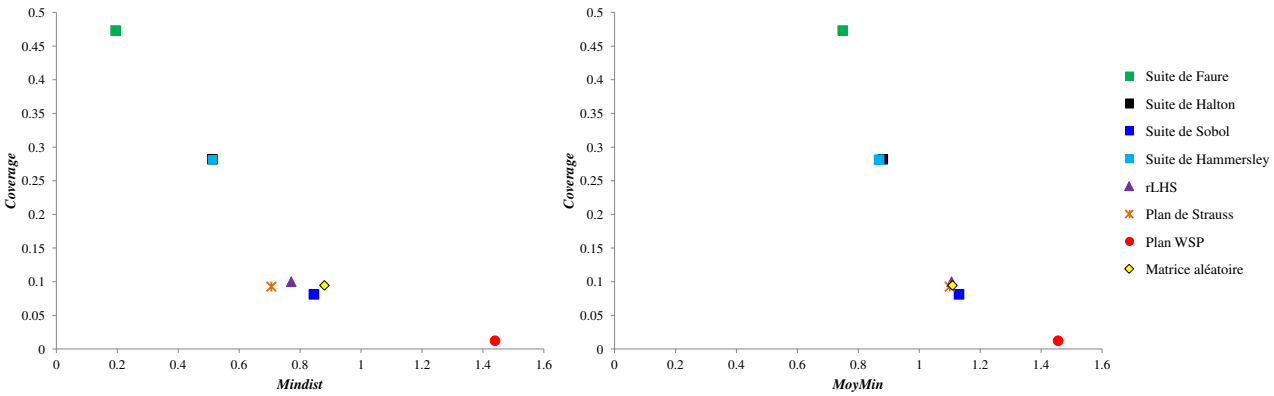


FIGURE 2.46 – Comparaison de la qualité intrinsèque de quelques SFD construits en 20D avec 400 points.

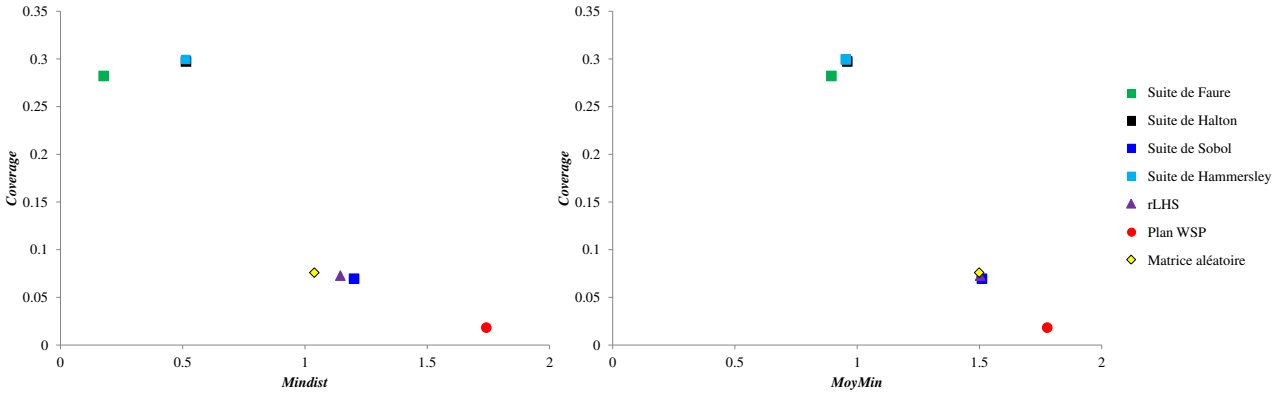


FIGURE 2.47 – Comparaison de la qualité intrinsèque de quelques SFD construits en 30D avec 600 points.

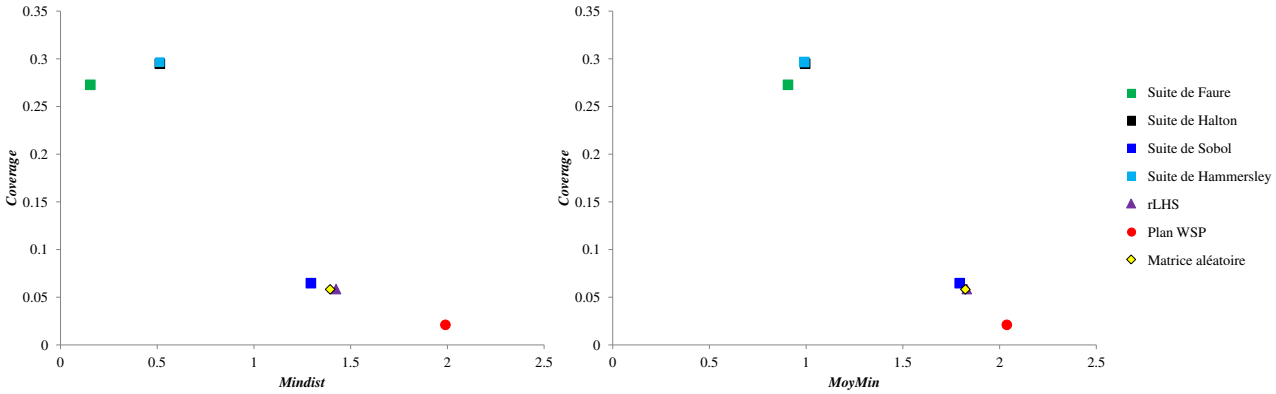


FIGURE 2.48 – Comparaison de la qualité intrinsèque de quelques SFD construits en 40D avec 800 points.

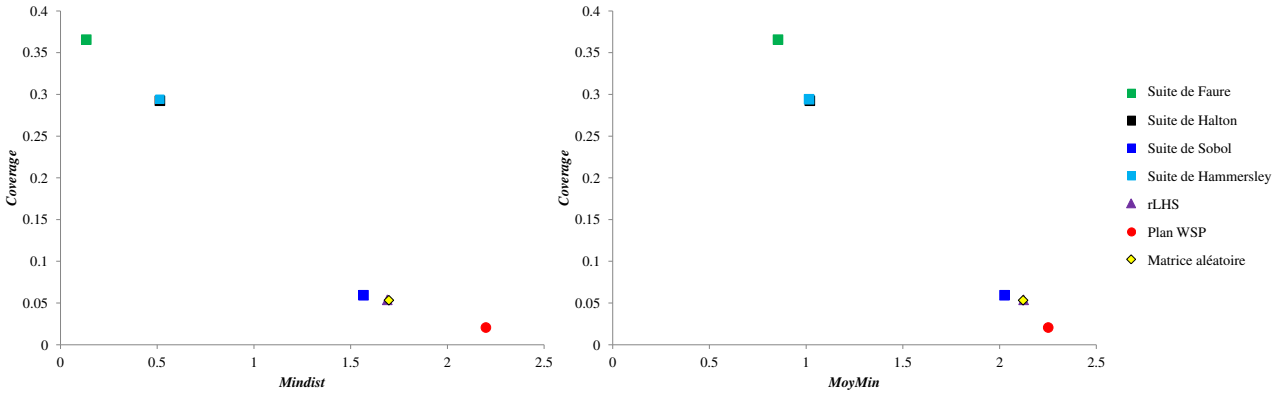


FIGURE 2.49 – Comparaison de la qualité intrinsèque de quelques SFD construits en 50D avec 1000 points.

Les figures 2.43 à 2.49 permettent de suivre l'évolution des critères intrinsèques des SFD lorsque la dimension augmente.

En 10D, la hiérarchisation des SFD change avec la suite de Faure qui est isolée sur les deux représentations graphiques avec la valeur *Coverage* la plus élevée. Un groupe avec de meilleurs critères

est constitué par le plan aléatoire, le plan rLHS et les suites de Sobol, Hammersley et Halton. Nous trouverons ensuite le plan de Strauss puis le plan WSP qui présente des critères bien meilleurs que les autres plans précités.

La comparaison des SFD en 20D montre de faibles valeurs *Mindist* et des valeurs *Coverage* élevées pour les suites de Faure, Halton et Hammersley alors que la suite de Sobol présente des critères d'uniformité similaires au plan rLHS, au plan de Strauss et à la matrice aléatoire. Le plan WSP se détache toujours des autres plans avec une valeur *Mindist* élevée et proche de la valeur *MoyMin* et une faible valeur *Coverage*.

En 30D, 40D et 50D, les plans de Strauss ne sont plus comparés car leur construction se révèle trop difficile et trop longue. Pour les autres plans, leur comportement est assez similaire à celui des plans en 20D avec néanmoins un regroupement plus marqué des trois suites à faible discrépance et une différenciation du plan WSP qui s'amoindrit quand la dimension augmente.

Nous venons de montrer que tous les plans uniformes ne sont pas équivalents en termes de critères de qualité lorsque la dimension augmente avec une différence notable dans la hiérarchisation des SFD à partir de 20 dimensions. Il nous faut donc résoudre les problèmes liés à la grande dimension et entre autres, disposer d'outils pour "réparer" les plans existants. En effet, des études ont montré que certains de ces plans ne présentent plus un recouvrement uniforme de l'espace avec des alignements, des accumulations de points ou des zones exemptes de points. Pour présenter les différentes étapes de "réparation" possibles, nous proposons d'étudier plus en détails quelques SFD en 20D.

3.2 Exemple en 20D

Pour cette étude, nous avons choisi de comparer différents types de plans avec 200 points, dont les comportements sont particuliers :

- un plan aléatoire,
- des suites à faible discrépance (suite de Sobol et suite de Faure), qui sont connues pour présenter des alignements, des lacunes ou des motifs,
- un plan avec des amas qui ont été générés par ajout de points très proches en trois zones d'un SFD classique.

Pour quantifier la qualité de ces plans, nous pouvons pour chacun d'entre eux calculer les critères classiques d'uniformité, c'est à dire les critères *Mindist*, *MoyMin* et *Coverage* dont les valeurs sont regroupées dans le tableau 2.12.

Tableau 2.12 – Valeurs *Mindist* et *Coverage* pour les plans étudiés en 20D avec 200 points.

20D - 200 points	*Mindist*	*MoyMin*	*Coverage*
Matrice aléatoire	0.867	1.175	0.094
Suite de Sobol	0.925	1.186	0.087
Suite de Faure	0.194	0.710	0.521
Plan avec amas	0.286	1.304	0.216

Ces valeurs montrent que ces plans ne sont pas tous équivalents en termes d'uniformité, ce qui se visualise aisément sur la figure 2.50. Entre autres, les faibles valeurs *Mindist* par rapport aux valeurs *MoyMin* pour certains plans indiquent une très grande proximité entre certains points.

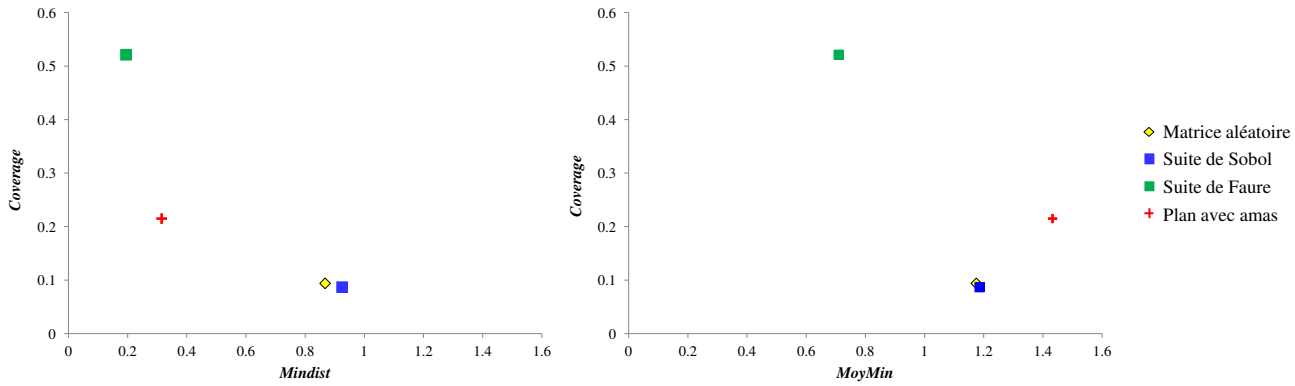


FIGURE 2.50 – Représentation graphique des critères d'uniformité des plans en 20D et 200 points.

Cette mauvaise répartition des points dans l'espace peut aussi être visualisée par des représentations graphiques sous la forme de plans de coupe. Ainsi, si nous représentons des plans de projections des suites de Faure et de Sobol (figure 2.51), nous observons des alignements, des lacunes et des motifs.

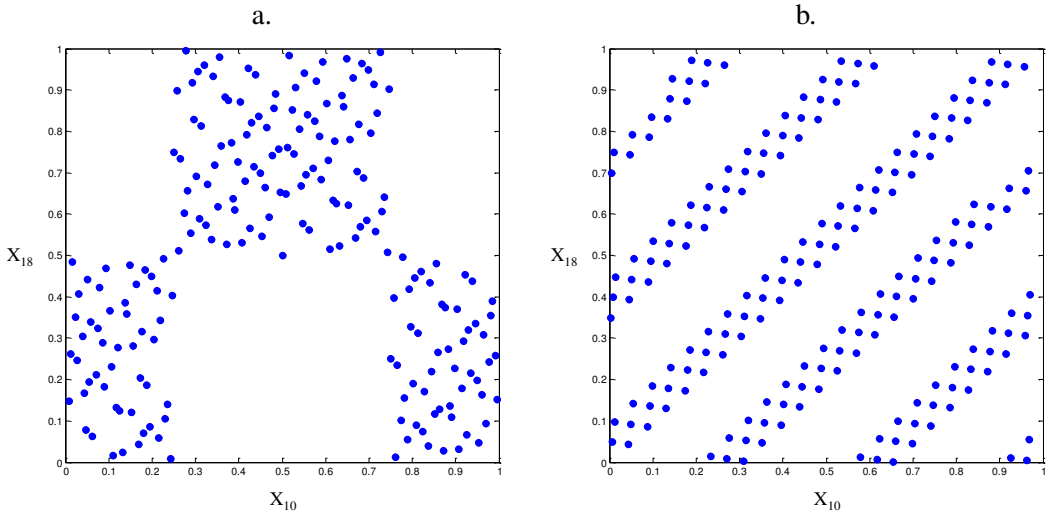


FIGURE 2.51 – Projection des points sur les axes factoriels (X_{10}, X_{18}) pour des suites à faible discrépance en 20 dimensions et 200 points pour **a)** une suite de Sobol et **b)** une suite de Faure.

De même, nous pouvons représenter les plans de coupe du plan avec amas (figure 2.52), les points constituant les amas étant représentés par des croix rouges, et assez surprenamment, nous ne retrouvons pas un regroupement des croix rouges.

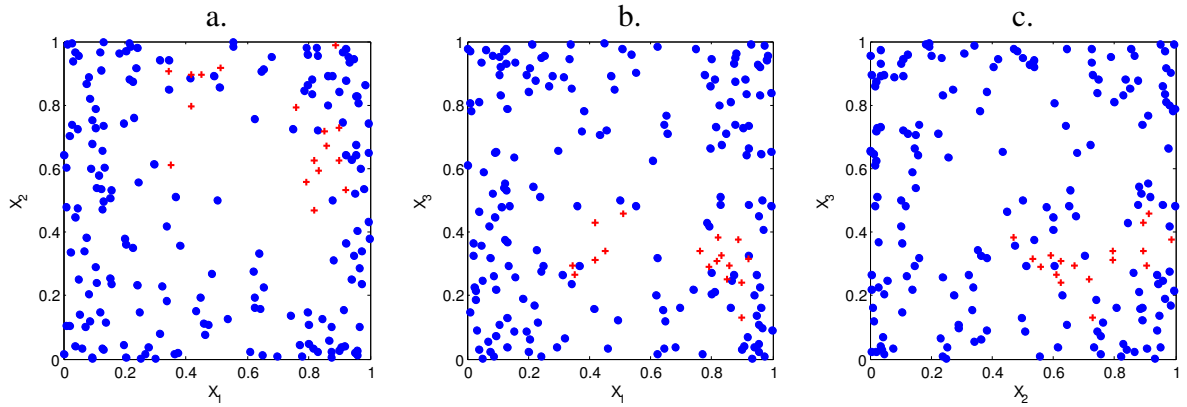


FIGURE 2.52 – Plan avec amas en 20 dimensions visualisé **a)** dans le plan (X_1, X_2), **b)** dans le plan (X_1, X_3) et **c)** dans le plan (X_2, X_3). Les croix rouges représentent les points constituant les amas générés par l'ajout de points très proches.

Ces observations nous amènent à penser que la connaissance *a priori* sur le mauvais conditionnement de certains plans à partir des critères intrinsèques n'est pas suffisante pour identifier réellement les "défauts" des plans. De plus, la grande dimension ne laisse aux simples représentations graphiques qu'un intérêt très limité pour visualiser ces répartitions de points car nous ne pouvons envisager que des plans de coupe (figure 2.52) en deux dimensions qui sont en nombre presque infini. Néanmoins, un outil graphique en deux dimensions demeurerait idéal. Aussi, nous avons envisagé d'utiliser l'une des méthodes de visualisation de données présentée dans la Partie 1 et plus précisément l'Analyse en Composantes Curvilignes. En effet, cette méthode nous a semblé pertinente dans la mesure où dans son principe de réduction de dimension, elle conserve les faibles distances. Ainsi, si nous appliquons l'ACC dans l'espace en 20 dimensions aux plans précités, nous obtenons une projection des points dans un espace en deux dimensions (figure 2.53), qui devrait permettre de visualiser rapidement la

présence éventuelle d'amas. L'application de l'ACC sur les 4 plans étudiés conduit aux représentations ci-dessous :

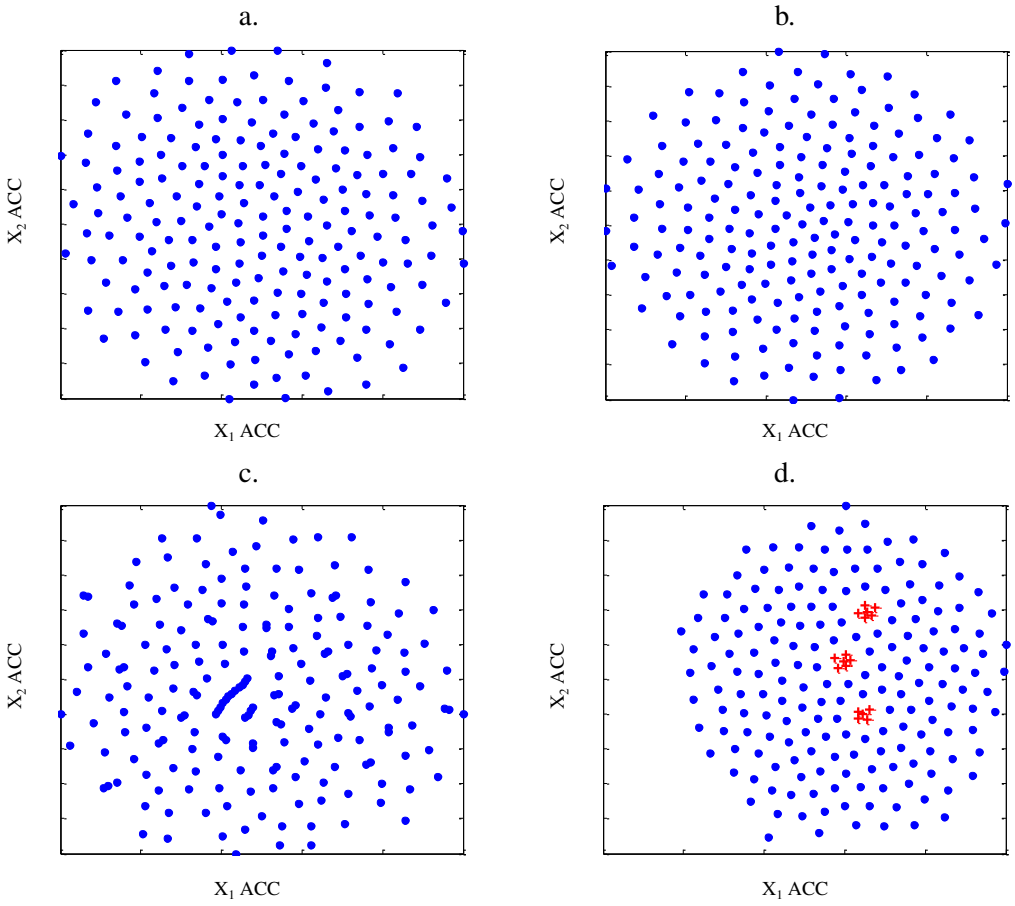


FIGURE 2.53 – Projections ACC de 4 plans en 20D et 200 points : **a)** une matrice aléatoire, **b)** la suite de Sobol, **c)** la suite de Faure, **d)** un plan avec amas. Tous les plans sont représentés dans l'espace de projection.

La figure 2.53 d. montre de manière évidente, le regroupement des points correspondant aux amas ajoutés volontairement, tout comme la figure 2.53 c. met en évidence des alignements de points connus dans les suites de Faure en grande dimension.

Pour compléter cette information visuelle qui, bien que très parlante, n'est pas encore suffisante, nous proposons de calculer d'autres indicateurs plus quantitatifs qui caractériseront la qualité de la distribution des points dans l'espace des composantes curvilignes. Ces indicateurs, présentés dans la Partie 1, sont les ratios R_{min} et R_{moy} calculés respectivement à partir des valeurs *Mindist* et *MoyMin* d'un plan uniforme de référence construit en 2D avec le même nombre de points.

Dans notre exemple, nous considérons respectivement les valeurs *Mindist* et *MoyMin* d'un plan uniforme de référence construit en 2D avec 200 points par l'algorithme WSP ($Mindist = MoyMin = 0.066$) que nous comparons aux valeurs de chaque plan après projection en composantes curvilignes (tableau 2.13).

Tableau 2.13 – Critères *Mindist* et *MoyMin* des plans après projection ACC pour calculer les ratios R_{min} et R_{moy}.

	Mindist après ACC	*MoyMin* après ACC	Plan de référence : 2D – 200 points *Mindist* = *MoyMin*	R_{min}	R_{moy}
Matrice aléatoire	0.045	0.059	0.066	**0.68**	**0.90**
Suite de Sobol	0.045	0.061		**0.68**	**0.93**
Suite de Faure	0.012	0.045		**0.18**	**0.69**
Plan avec amas	0.015	0.061		**0.22**	**0.93**

Les valeurs regroupées dans le tableau 2.13, montrent une grande variation des valeurs R_{min}. La suite de Faure et le plan avec amas présentent des ratios R_{min} faibles (respectivement 0.18 et 0.22) confirmant la présence d'amas qui ont été visualisés dans la représentation graphique de l'ACC (figure 2.53 c. et d.). Toutefois, les valeurs R_{moy} ne sont pas équivalentes, avec $R_{moy} = 0.69$ pour la suite de Faure et $R_{moy} = 0.93$ pour le plan avec amas, signifiant que ce dernier est globalement mieux conditionné que la suite de Faure. Cette dernière observation va dans le même sens que la construction même du plan avec amas qui, nous le rappelons, résulte de l'ajout de points très proches à un plan uniforme classique. En effet, la faible valeur $R_{min} = 0.22$ caractérise la présence de points très proches et la valeur $R_{moy} = 0.93$ indique que la distribution des distances minimales est proche de celle de la matrice de référence. Ainsi, si nous venions à éliminer les points ajoutés pour former les amas nous obtiendrions une répartition presque uniforme.

Cette comparaison des ratios démontre l'importance d'interpréter simultanément les valeurs R_{min} et R_{moy} pour obtenir des informations concernant à la fois la présence de points très proches, par l'interprétation du R_{min} et le conditionnement global du plan, par l'interprétation du R_{moy}. Dans cette étude, les ratios R_{min} et R_{moy} montrent la présence d'amas et/ou de lacunes, ce qui nous a conduit à nous interroger sur les possibilités de réparation, à savoir l'élimination des amas si nécessaire et le comblement des zones lacunaires, qui peuvent être très pénalisantes pour les étapes de modélisation.

3.2.1 Réparation des plans

3.2.1.1 Étape 1 : Élimination des amas

Pour éliminer les amas, il nous faut construire un plan de référence en 20D et 200 points pour connaitre la valeur d_{min} de référence qui sera utilisée ensuite pour détecter les éventuels amas et supprimer tous les points distants d'une valeur plus faible que la valeur d_{min} de référence. Dans notre cas, la distance minimale entre deux points du plan de référence est égale à 1.487 ce qui définit la valeur d_{min} de référence. Le tableau 2.14 présente les valeurs des ratios R_{min} et R_{moy} avant et après réparation des SFD ainsi que l'évolution du nombre de points au cours de ces étapes successives.

Tableau 2.14 – Comparaison des ratios R_{min} et R_{moy} avant et après réparation des plans en 20D.

	Ratios des plans initiaux		Nombre de points		Après réparation totale		*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratios des plans réparés	
	R_{min}	R_{moy}	Après élimination des amas	Après remplissage des lacunes	*Mindist*	*MoyMin*		R_{min}	R_{moy}
Matrice aléatoire	**0.68**	**0.90**	42	83	0.088	0.1	0.101	**0.88**	**0.98**
Suite de Sobol	**0.68**	**0.93**	6	21	0.189	0.221	0.232	**0.81**	**0.95**
Suite de Faure	**0.18**	**0.69**	3	25	0.172	0.186	0.196	**0.88**	**0.95**
Plan avec amas	**0.22**	**0.93**	186	186	0.056	0.065	0.067	**0.84**	**0.95**

Les valeurs reportées dans le tableau 2.14, nous montrent que la suppression des points très proches pour le plan avec amas, conduit à la suppression de 14 points, qui sont en fait les 14 points ajoutés volontairement à un SFD à 186 points.

Nous observons aussi que le nombre de points retenus après suppression des amas est très faible pour les suites à faible discrépance ce qui signifie que par construction ces plans présentent des accumulations de points dans certaines zones de l'espace des variables.

Par ailleurs, la matrice aléatoire et la suite de Sobol, qui présentent des ratios R_{min} et R_{moy} initiaux équivalents, comptent respectivement 42 et 6 points après suppression des amas, alors que les projections ACC (figures 2.53 a. et b.) ne montrent pas de points très proches. Pour mieux comprendre ce phénomène, nous proposons de comparer les critères *Mindist* et *MoyMin* des plans initiaux (tableau 2.12) aux valeurs du plan de référence en 20D et 200 points ($Mindist = 1.487$ et $MoyMin = 1.511$) : le critère *Mindist* de la matrice aléatoire et de la suite de Sobol est plus faible que celui du plan de référence, ce qui signifie que les points sont à une distance inférieure à la valeur d_{min} de référence sans pour autant former un amas.

3.2.1.2 Étape 2 : Remplissage des lacunes

Pour cette deuxième étape de réparation, nous considérons les plans après avoir éliminé les points trop proches à savoir la matrice aléatoire à 42 points, la suite de Sobol à 6 points, la suite de Faure à 3 points et le plan avec amas à 186 points. Pour chacun de ces plans nous ajoutons le même plan candidat à 10000 points puis nous comblons les lacunes selon la procédure décrite dans la première partie de ce manuscrit.

Si nous considérons le plan avec amas, nous constatons que cette étape n'ajoute pas de points. Ceci s'explique par le fait qu'après la première étape d'élimination des amas nous avons retrouvé le plan uniforme à 186 points que nous avions utilisé initialement pour ajouter les amas.

Pour les 3 autres plans, cette étape conduit à l'ajout d'une vingtaine de points pour les plans qui présentaient de nombreux amas, comme les suites à faible discrépance, et d'une quarantaine de points pour le plan aléatoire.

De manière générale, si la distribution de points présente des zones de vide ou lacunes, nous pourrons alors les combler selon cette procédure.

3.2.1.3 Étape 3 : Application de l'ACC et calcul des ratios R

Après avoir supprimé les amas puis comblé les lacunes, nous projetons par ACC les différents plans "reconstruits" et nous calculons les critères *Mindist* et *MoyMin* dans l'espace de projection en 2D. Pour chaque plan, nous calculons les nouvelles valeurs des ratios R_{min} et R_{moy} en considérant les plans après réparation totale. Pour ce faire, nous construisons pour chaque plan, un plan de référence en 2D pour le nombre de points constituant le plan étudié après réparation totale. Dans chaque cas nous calculerons alors les ratios des critères *Mindist* et *MoyMin* des SFD "réparés" avec les critères du plan de référence correspondant.

Les résultats du tableau 2.14 méritent d'être commentés. Tout d'abord, nous observons que tous les plans réparés présentent des ratios R_{min} et R_{moy} élevés dont les valeurs sont respectivement proches de 0.80 et 0.95 ce qui garantit une distribution uniforme des points dans l'espace. Ainsi, les plans tels que la suite de Faure et le plan avec amas, qui présentaient initialement de faibles valeurs R_{min} caractérisant la présence d'amas, peuvent être réparés par cette méthode ce qui explique une augmentation des ratios vers la valeur 1. Cependant, les ratios obtenus pour les suites de Sobol et Faure doivent être interprétés avec précaution car après la suppression des amas, seulement 6 et 3 points ont été conservés. Lors de la deuxième étape de réparation, seulement 21 et 25 points sont ajoutés par l'algorithme WSP pour garantir une bonne uniformité, mais il semble évident que l'espace sera moins bien rempli puisque le nombre de points est moindre. L'interprétation des ratios, qui renseigne sur l'uniformité de la répartition des points, doit donc toujours être complétée en considérant le nombre de points.

3.2.2 Conclusion

Dans cet exemple, nous avons choisi de nous intéresser à des SFD en grande dimension (20D) qui peuvent présenter des zones très denses et/ou des zones de vide. Nous avons utilisé l'algorithme WSP pour "réparer" ces plans mal conditionnés et l'ACC pour visualiser les points dans un espace en 2D et pour calculer les ratios R_{min} et R_{moy} qui complèteront l'interprétation visuelle. Parmi les 4 plans étudiés en 20D, nous avons observé des comportements différents lors des étapes de réparation ce qui signifie probablement que la nature des plans a son importance mais aussi que la notion de réparation se complique avec la dimension.

3.3 Repliage

Tout comme dans le contexte de l'expérimentation, l'analyse de sensibilité en simulation numérique cherche à étudier l'impact de la variation des facteurs d'entrée du modèle X_i sur la variable de sortie (réponse Y). Autrement dit, nous cherchons à identifier les paramètres ou les combinaisons de paramètres qui contribuent le plus à la variabilité de la réponse du modèle que nous qualifierons de facteurs influents. Cette information est primordiale car elle va permettre de simplifier le modèle et de réaliser des études plus fines, du type de surfaces de réponse en se focalisant sur les variables influentes. De nombreuses méthodes d'analyse de sensibilité existent [102, 103, 104, 88, 105] mais elles nécessitent des plans d'expériences spécifiques à la méthode qui ne peuvent pas toujours être utilisés par la suite. Or, l'analyse de sensibilité n'est souvent que la première étape d'une étude se terminant par l'ajustement d'un métamodèle (modèle de modèle) plus simple mais réaliste. Cette deuxième étape nécessitera alors de nouvelles simulations dans l'espace des variables influentes en construisant généralement un plan uniforme. Il faut se souvenir que même si nous parlons d'expériences numériques, les temps de calcul peuvent être très longs et il est alors impératif d'établir des stratégies économes en termes de simulations.

L'analyse de sensibilité va permettre d'extraire le sous-ensemble de D' facteurs influents qui constituera le sous-espace de travail de la suite de l'étude et il serait intéressant de conserver les simulations déjà effectuées. Cette étape appelée "repliage" va consister à projeter les points du plan d'expériences initial de dimension D dans la sous-espace des variables influentes de dimension D'. Cette étape de réduction de dimension peut générer dans l'espace d'arrivée des superpositions de points, synonyme d'amas, ou des zones lacunaires, déficientes en information. On peut alors se demander si cette dégradation de l'uniformité de la distribution de points dans le nouvel espace d'intérêt en D' dimensions est préjudiciable ou non. Pour répondre à cette question, nous proposons d'étudier l'impact du "repliage" sur la qualité des plans en dimension D' en fonction du type de matrice initiale de dimension D et du nombre de facteurs conservés. Nous proposons une démarche en plusieurs étapes du repliage à la réparation du plan, que nous appliquerons en 10, 30 et 50 dimensions.

3.3.1 Démarche

Pour cette étude, nous avons choisi différents plans initiaux et pour chacun de ces plans en 10, 30 et 50 dimensions, nous avons effectué un "repliage" dans des conditions plus ou moins sévères, à savoir ne retenir que peu de facteurs (environ 25%) ou la moitié des facteurs ou un grand nombre (environ 75%). Le calcul des critères d'uniformité nous permettra d'évaluer les conséquences du "repliage" sur la qualité des plans résultants. Cette étape de repliage risquant de favoriser la création d'amas et/ou de lacunes dans le sous-espace, nous visualiserons d'abord la répartition des points par l'analyse en composantes curvilignes (ACC), puis nous réparerons les plans si besoin.

En résumé, l'étude se déroulera de la façon suivante :

- Calcul des critères intrinsèques des plans en dimension D,
- "Repliage" dans l'espace de dimension D', avec $D' < D$,
- Calcul des critères intrinsèques des plans repliés,
- Détection des amas par ACC,
- Suppression des amas par l'algorithme WSP,
- Remplissage des lacunes par l'algorithme WSP,

– Calcul des ratios R sur les plans repliés et réparés.

Pour chaque étape, nous proposons de représenter graphiquement l'étude des plans en 10D repliés sur peu de facteurs, $D' = 2$, afin de suivre les étapes du repliage, d'élimination et/ou d'ajout de points.

3.3.2 Caractérisation des plans en dimension D

Nous avons considéré des matrices aléatoires et des SFD classiques tels que des suites à faible discrépance (suites de Faure et suites de Sobol), des hypercubes latins aléatoires (rLHS) et des plans issus de l'algorithme WSP en 10, 30 et 50 dimensions avec deux points par dimension. Pour chaque plan et pour chaque dimension, nous avons calculé les valeurs *Mindist*, *MoyMin* et *Coverage* ainsi que les ratios R_{min} et R_{moy} à partir des projections ACC de ces plans initiaux. Toutes ces valeurs sont regroupées dans le tableau 2.15.

Tableau 2.15 – Critères des plans initiaux en 10, 30 et 50 dimensions avec 2 points par dimension.

		Mindist	***MoyMin***	***Coverage***	R_{min}	R_{moy}
10D - 20 points	Matrice aléatoire	0.6252	0.7982	0.1101	0.69	0.88
	Suite de Sobol	0.6917	0.8266	0.109	0.73	0.94
	Suite de Faure	0.2875	0.5441	0.5782	0.39	0.72
	rLHS	0.6859	0.8599	0.1236	0.75	0.92
	Plan WSP	1.2302	1.2536	0.0195	0.79	0.94
30D - 60 points	Matrice aléatoire	1.3593	1.6732	0.0721	0.8	0.97
	Suite de Sobol	1.2482	1.5528	0.0925	0.68	0.97
	Suite de Faure	0.1767	0.5531	0.7275	0.18	0.56
	rLHS	1.4542	1.7078	0.0622	0.85	1.02
	Plan WSP	1.919	1.947	0.0107	0.86	1.03
50D - 100 points	Matrice aléatoire	1.9006	2.2716	0.0555	0.75	0.95
	Suite de Sobol	1.7336	2.0216	0.0662	0.76	1
	Suite de Faure	0.1334	0.5103	0.8332	0.1	0.51
	rLHS	1.9338	2.3074	0.0597	0.79	0.97
	Plan WSP	2.393	2.4208	0.0087	0.82	0.97

Pour faciliter la comparaison de ces plans, nous proposons de représenter les critères intrinsèques à l'aide des graphes : $Coverage = f(Mindist)$ et $Coverage = f(MoyMin)$.

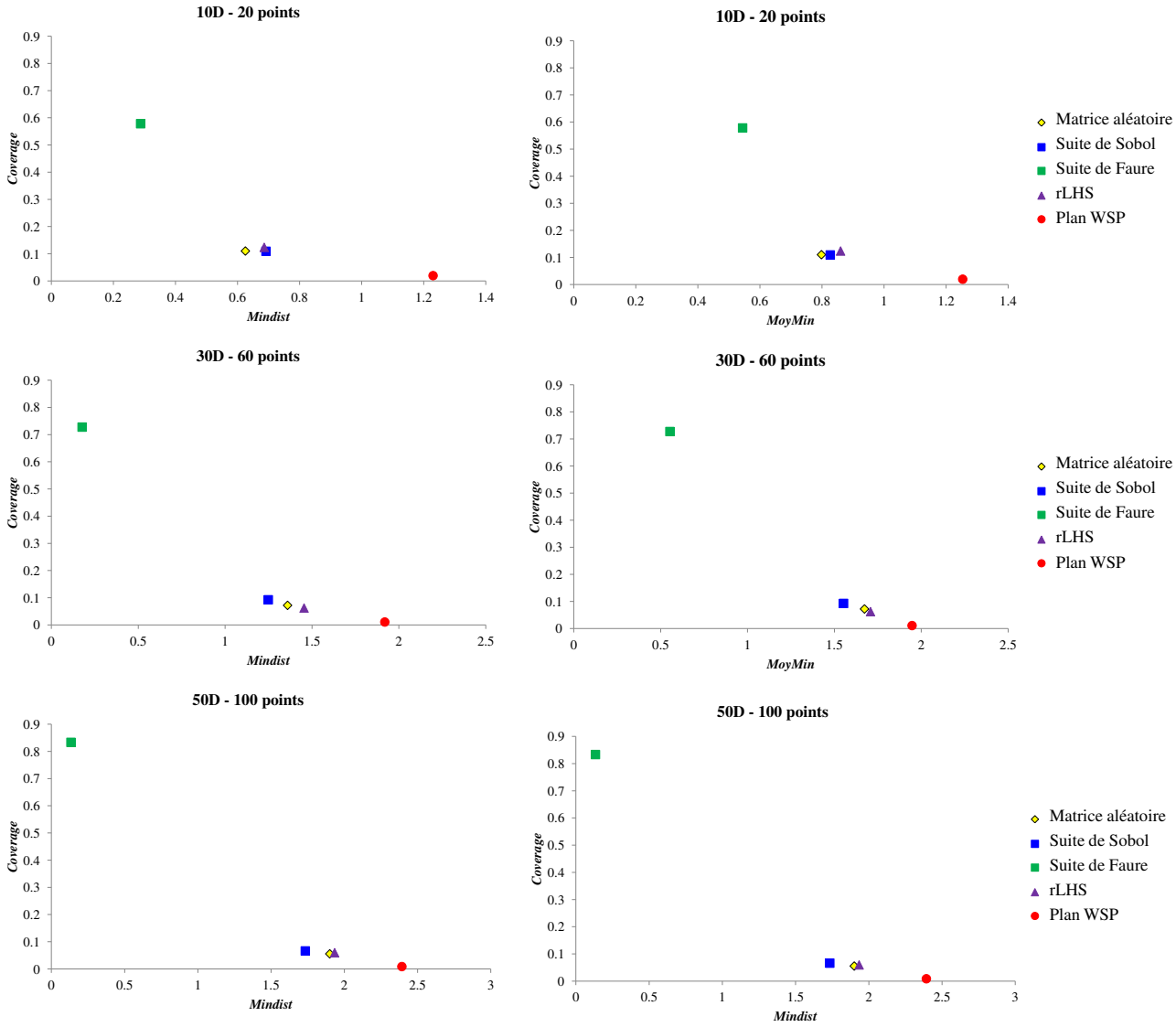


FIGURE 2.54 – Comparaison des critères intrinsèques de qualité des plans en D dimensions.

La comparaison de la qualité intrinsèque des plans par la représentation graphique des valeurs *Coverage* en fonction des valeurs *Mindist* ou *MoyMin* met en évidence trois groupes de plans quelle que soit la dimension étudiée. Tout d'abord, les suite de Faure présentent les plus mauvais critères avec une valeur *Coverage* élevée et de faibles valeurs *Mindist* et *MoyMin*, traduisant une mauvaise distribution des points dans l'espace avec la présence de lacunes et/ou d'amas. Les suites de Sobol, les plans rLHS et les matrices aléatoires constituent le deuxième groupe avec des valeurs *Mindist* et *MoyMin* plus élevées que celles de la suite de Faure et des valeurs *Coverage* très faibles. Nous notons qu'une simple construction aléatoire conduit à des critères équivalents à ceux résultant de certains algorithmes de construction. Le dernier groupe est constitué par les plans WSP qui présentent les meilleurs critères avec les valeurs *Mindist* et *MoyMin* les plus élevées et *Coverage* les plus faibles. Nous rappelons qu'une valeur *MoyMin* proche du *Mindist*, est synonyme d'une répartition uniforme des points dans l'espace.

3.3.3 Repliage

Le principe de l'étape de repliage, qui consiste en une réduction de dimension, peut être schématisé ainsi (figure 2.55).

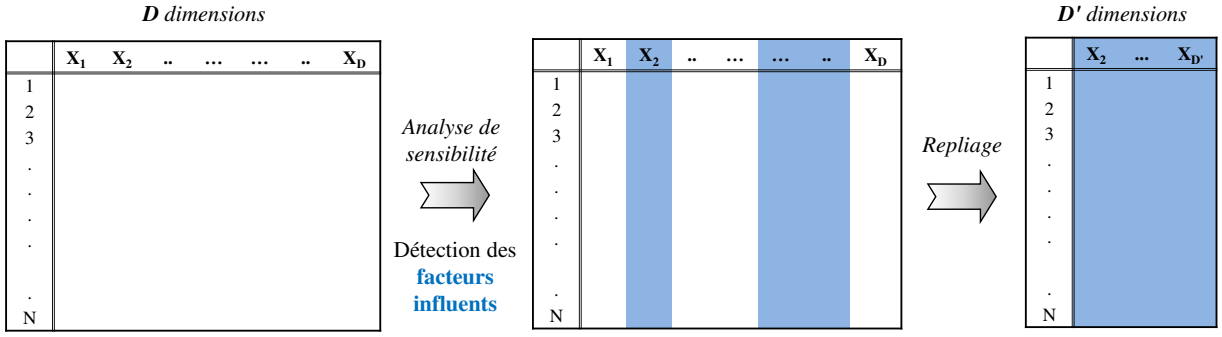


FIGURE 2.55 – Illustration du "repliage".

Pour cette étude nous avons choisi aléatoirement les facteurs conservés dans les sous-espaces pour chaque pourcentage de facteurs projetés, pour chaque plan et pour toutes les dimensions. Par exemple, lorsque nous replions les suites de Sobol en 10D dans un sous-espace en 2D, nous conservons les facteurs X_5 et X_8 alors que pour le repliage sur la moitié des facteurs nous conservons les facteurs X_2, X_3, X_4, X_6 et X_9.

Pour comparer la qualité des plans après repliage, nous calculerons leurs critères intrinsèques mais pour juger leur qualité "en absolue", nous avons besoin de plans de référence construits en même dimension avec le même nombre de points que le plan considéré après "repliage". Ces plans de référence sont construits par l'algorithme de sélection WSP et leurs critères sont présentés dans le tableau 2.16.

Tableau 2.16 – Critères des plans de référence.

	Critères des plans de référence	***Mindist***	***MoyMin***	***Coverage***
Repliage des plans en 10D	**2D – 20 points**	0.23	0.2304	0.0019
	5D – 20 points	0.8441	0.8527	0.0218
	7D – 20 points	1.06	1.0839	0.0222
Repliage des plans en 30D	**7D – 60 points**	0.8060	0.8112	0.0112
	15D – 60 points	1.3970	1.4229	0.0164
	22D – 60 points	1.644	1.6709	0.0163
Repliage des plans en 50D	**12D – 100 points**	1.165	1.1786	0.0139
	25D – 100 points	1.7184	1.7455	0.0139
	37D – 100 points	2.0356	2.0677	0.0162

3.3.4 Caractérisation des plans en dimension D'

Précédemment, nous avons montré que les critères intrinsèques des plans en D dimensions ($D = 10\,;\,30$ ou 50) les discriminaient en trois groupes : des plans de mauvaise qualité, des plans de qualité intermédiaire et le plan WSP qui présente la meilleure répartition des points. Nous souhaitons alors évaluer les conséquences du repliage sur les propriétés des plans en dimension D'.

3.3.4.1 Caractérisation des plans 10D repliés

A partir des plans en 10D, nous proposons de conserver une majorité (7D), la moitié (5D) et peu de facteurs (2D) afin de vérifier si le repliage conserve le classement des plans en fonction de leur qualité intrinsèque.

Dans le tableau 2.17, nous regroupons les valeurs des critères intrinsèques des plans repliés.

Tableau 2.17 – Repliage des plans en **10D** dans des sous-espaces en **7D**, **5D** et **2D**.

Plans initiaux: 10D – 20 points		***Mindist***	***MoyMin***	***Coverage***	R_{min}	R_{moy}
Repliage **7D**	Matrice aléatoire	0.2605	0.5444	0.2179	0.39	0.88
	Suite de Sobol	0.4667	0.6464	0.1720	0.74	0.92
	Suite de Faure	0.2405	0.3620	0.6738	0.39	0.59
	rLHS	0.4008	0.6529	0.1711	0.52	0.87
	Plan WSP	0.3610	0.8789	0.2251	0.34	0.83
Repliage **5D**	Matrice aléatoire	0.2556	0.3949	0.2197	0.51	0.78
	Suite de Sobol	0.4507	0.5197	0.1034	0.68	0.85
	Suite de Faure	0.2033	0.2033	$3.27e^{-9}$	0.52	0.52
	rLHS	0.2935	0.4484	0.2673	0.53	0.77
	Plan WSP	0.1952	0.6159	0.3227	0.24	0.84
Repliage **2D**	Matrice aléatoire	0.0471	0.1168	0.5750	0.19	0.55
	Suite de Sobol	0.0884	0.1534	0.4644	0.46	0.76
	Suite de Faure	0.1286	0.1286	$3.3e^{-9}$	0.58	0.58
	rLHS	0.0688	0.1495	0.3751	0.27	0.63
	Plan WSP	0.0195	0.1144	0.7470	0.07	0.39

A partir de ces critères, nous pouvons clairement observer que la qualité des plans varie en fonction du pourcentage de facteurs conservés. Nous proposons de faciliter cette interprétation par les graphes de la figure 2.56.

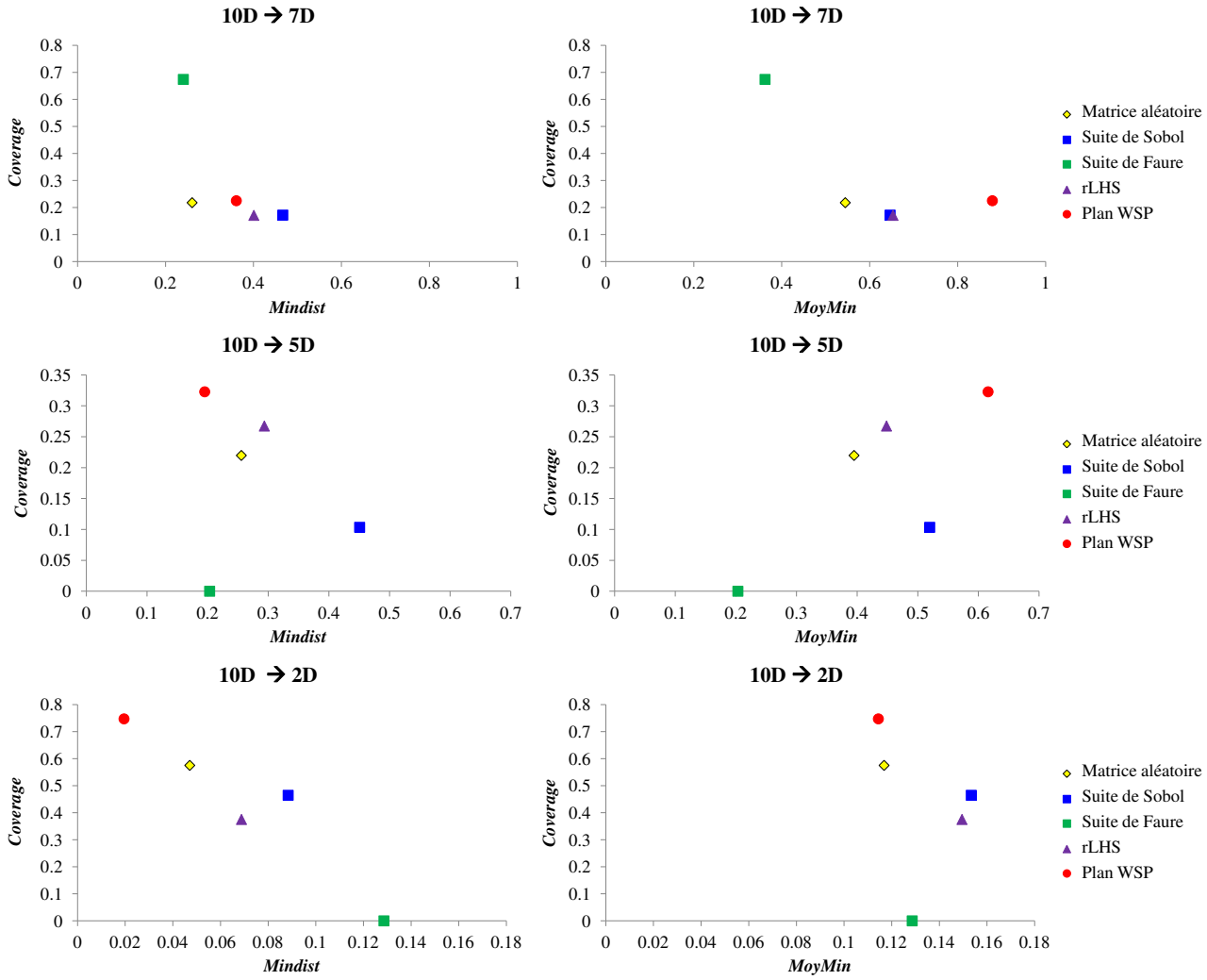


FIGURE 2.56 – Évolution des critères intrinsèques de qualité des plans 10D repliés.

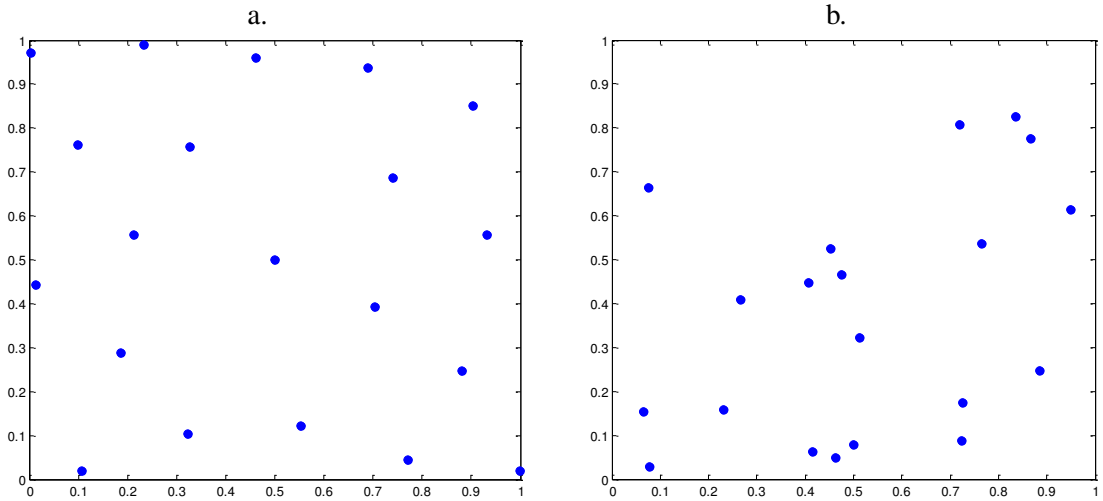
Le tableau 2.17 et la figure 2.56 montrent qu'en 10D les plans repliés ne sont pas tous équivalents en termes de qualité. Cette observation est valable quel que soit le type de plan pour un nombre fixé de facteurs mais aussi pour différents pourcentages de facteurs repliés. De plus, nous observons sur les graphes *Coverage* en fonction du critère *Mindist* ou *MoyMin* que la position des plans repliés est différente pour un nombre donné de facteurs, ce qui signifie qu'il faut distinguer le comportement local (critère *Mindist*) du conditionnement global (critère *MoyMin*) d'un plan.

Si nous considérons un repliage conservant la majorité des facteurs ($10D \rightarrow 7D$), la suite de Faure présente une valeur *Mindist* égale à 0.2405 et une valeur *MoyMin* égale à 0.3620 alors que la matrice aléatoire qui présente une valeur *Mindist* similaire ($= 0.2605$) a une valeur *MoyMin* plus élevée ($= 0.5444$) qui traduit un meilleur conditionnement global. La suite de Sobol et le plan rLHS présentent des critères très proches et donc une qualité intrinsèque équivalente avec un bon conditionnement général. La valeur *MoyMin* la plus élevée ($= 0.8789$) est obtenue pour le plan WSP et reste plus importante que la valeur *Mindist* ($= 0.3610$) signifiant que cette répartition des points est la plus uniforme parmi les plans étudiés mais avec des points qui sont très proches dans le nouvel espace d'intérêt.

Lorsque nous ne conservons que la moitié des facteurs ($10D \rightarrow 5D$), la matrice aléatoire et le plan rLHS présentent des valeurs de critères équivalentes avec des valeurs *Mindist* respectivement égales à 0.2556 et 0.2935 et *MoyMin* égales à 0.3949 et 0.4484. Les valeurs *Mindist* et *MoyMin* comparables permet d'éliminer l'hypothèse de la présence d'amas de points. La matrice WSP présente une faible valeur *Mindist* ($= 0.1952$) et une valeur *MoyMin* élevée ($= 0.6159$) ce qui traduit un bon conditionnement global mais avec la présence d'au moins deux points très proches. Les critères intrinsèques de la suite de Faure, qui étaient initialement les plus mauvais, se déplacent dans le positionnement relatif lors du repliage. En effet, les valeurs *Mindist* et *MoyMin* sont égales à 0.2033 ce qui signifie que tous les points sont à une même distance de leur plus proche voisin et la valeur *Coverage* est proche de zéro. Ce phénomène est la conséquence des alignements de points dans l'espace des variables avec une distance égale à 0.2033 entre chaque point et son plus proche voisin mais nous ne pouvons pas conclure sur le bon remplissage de l'espace.

Si nous considérons un repliage plus sévère avec la conservation de deux facteurs uniquement ($10D \rightarrow 2D$), quelle que soit la nature du plan les valeurs *MoyMin* sont équivalentes et de l'ordre de 0.12. Si nous comparons ces valeurs *MoyMin* aux valeurs *Mindist* associées, nous observons que pour certains plans comme la matrice aléatoire ($Mindist = 0.0471$), rLHS ($Mindist = 0.0688$) et le plan WSP ($Mindist = 0.0195$) les valeurs *Mindist* sont inférieures aux valeurs *MoyMin,* ce qui traduit la présence de deux ou plusieurs points très proches dans l'espace des variables. La comparaison des valeurs *Coverage* (0.5750 pour la matrice aléatoire, 0.3751 pour le plan rLHS et 0.7470 pour le plan WSP) nous permet de détecter la présence de lacunes. Le plan WSP présente la valeur *Coverage* la plus élevée synonyme de lacunes qui semblent être plus importantes que pour la matrice aléatoire. Par ailleurs, la suite de Faure qui présente initialement les plus mauvais critères initiaux ne conduit pas au plus mauvais plan replié avec des valeurs *Mindist* et *MoyMin* égales à 0.1286 et une valeur *Coverage* très faible ($= 3.3.10^{-9}$) qui s'explique par des alignements de points dans l'espace comme lors du repliage de la moitié des facteurs.

Le repliage des plans en 10D en ne conservant que 2 facteurs nous permet de représenter graphiquement les repartions de points lors des différentes étapes de cette étude (figure 2.57).



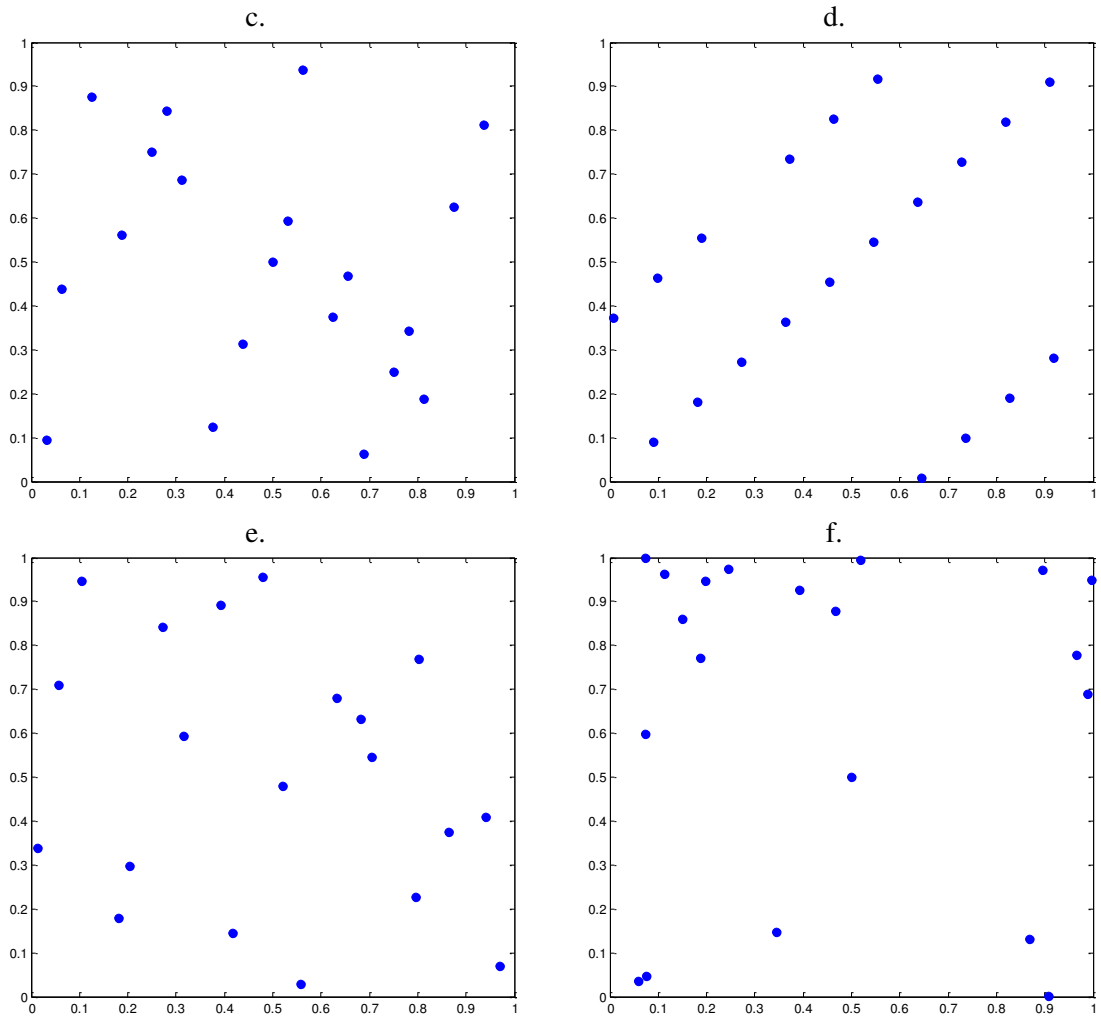


FIGURE 2.57 – Représentation des plans en 10D et 20 points repliés sur 2 facteurs que nous comparons au **a)** plan de référence en 2D et 20 points. **b)** Matrice aléatoire, **c)** suite de Sobol, **d)** suite de Faure, **e)** rLHS, **f)** plan WSP.

Sur ces représentations graphiques, nous observons que les répartitions des points ne sont pas toutes uniformes, certaines présentent des amas, d'autres des lacunes ou des alignements.

Ces mauvais conditionnements nous mèneront à envisager des stratégies pour rendre ces distributions de points les plus uniformes possibles.

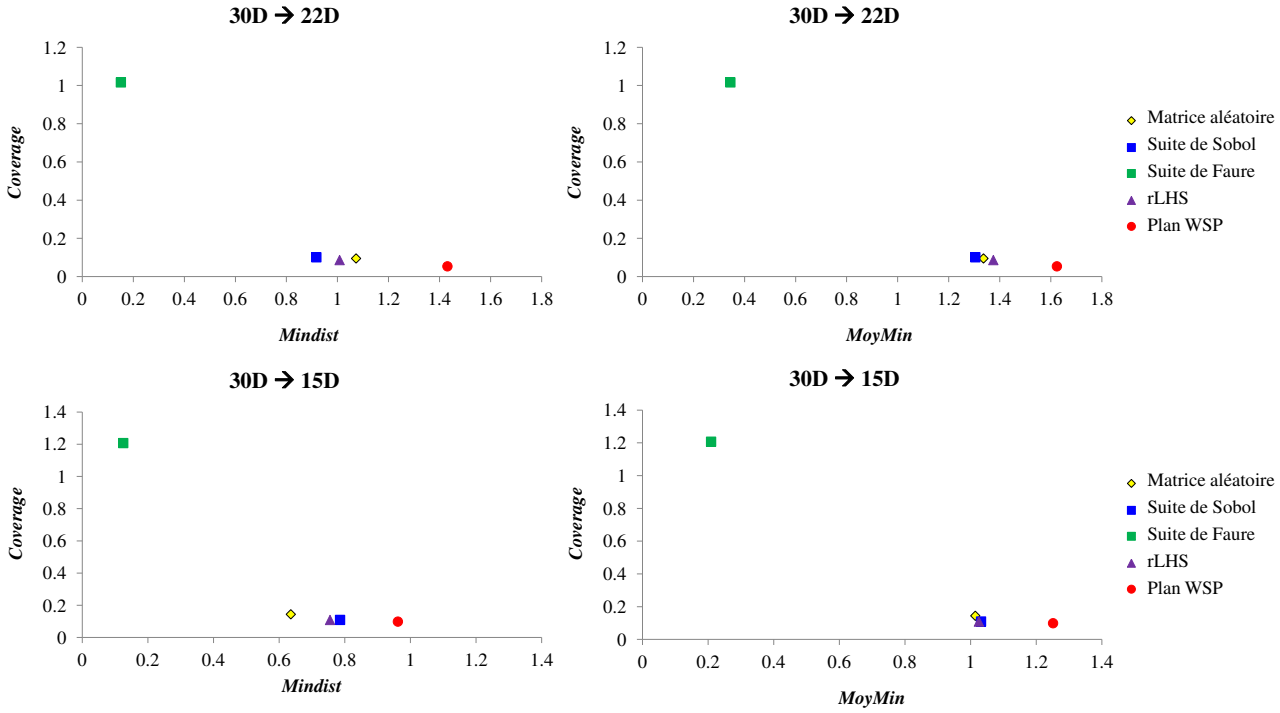
3.3.4.2 Caractérisation des plans 30D repliés

Les valeurs des critères intrinsèques de qualité des plans 30D repliés dans un sous-espace en 22D, 15D ou 7D sont regroupées dans le tableau 2.18.

Tableau 2.18 – Repliage des plans en **30D** dans des sous-espaces en **22D**, **15D** et **7D**.

Plans initiaux : 30D – 60 points		***Mindist***	***MoyMin***	***Coverage***	R_{min}	R_{moy}
Repliage **22D**	Matrice aléatoire	1.0728	1.3361	0.0950	0.75	0.95
	Suite de Sobol	0.9172	1.3042	0.1011	0.65	0.97
	Suite de Faure	0.1513	0.3444	1.0164	0.16	0.45
	rLHS	1.0083	1.3747	0.0865	0.70	0.97
	PlanWSP	1.4307	1.6237	0.0537	0.76	0.95
Repliage **15D**	Matrice aléatoire	0.6355	1.0142	0.1442	0.59	0.93
	Suite de Sobol	0.7858	1.0317	0.1093	0.72	0.97
	Suite de Faure	0.1249	0.2091	1.2071	0.16	0.39
	rLHS	0.7550	1.0252	0.1091	0.68	0.93
	PlanWSP	0.9618	1.2512	0.0988	0.69	0.95
Repliage **7D**	Matrice aléatoire	0.2632	0.4778	0.2024	0.45	0.81
	Suite de Sobol	0.2480	0.3936	0.2216	0.53	0.79
	Suite de Faure	0.0853	0.0853	$6.8e^{-9}$	0.26	0.26
	rLHS	0.2520	0.4891	0.2142	0.46	0.85
	PlanWSP	0.2970	0.5823	0.2711	0.45	0.85

La figure 2.58 représente graphiquement ces critères.



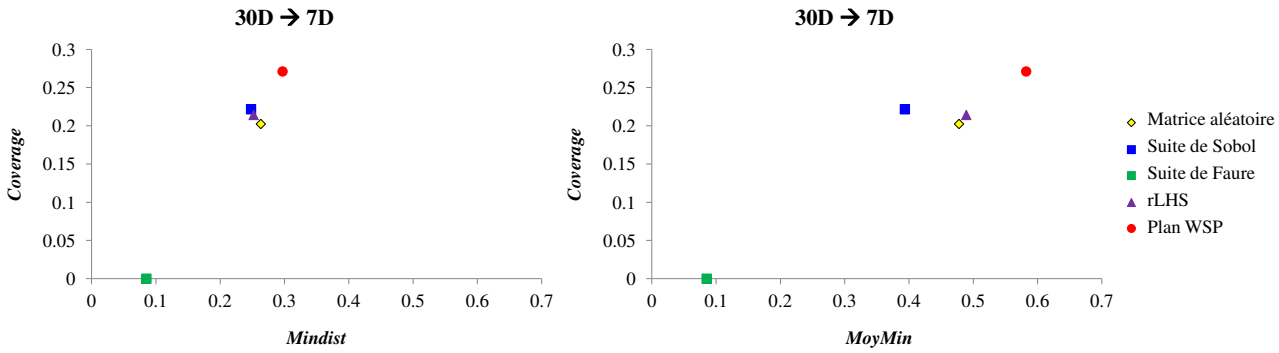


FIGURE 2.58 – Évolution des critères intrinsèques de qualité des plans 30D repliés.

L'étude du repliage des plans en 30D conduit à des résultats différents de ceux obtenus pour le repliage des plans en 10D. En effet, si on replie les plans en 30D sur une majorité de facteurs ($30D \rightarrow 22D$) nous observons que la suite de Faure présente toujours de mauvais critères intrinsèques avec une faible valeur *Mindist* ($= 0.1513$) qui est inférieure à la valeur *MoyMin* ($= 0.3444$) et une valeur *Coverage* élevée ($= 1.0164$) qui traduit à la fois la présence de points très proches et de lacunes dans l'espace des variables. La matrice aléatoire, la suite de Sobol et le plan rLHS présentent des valeurs *MoyMin* équivalentes (respectivement 1.3361, 1.3042, 1.3747) mais des valeurs *Mindist* plus faibles (respectivement 1.0728, 0.9172, 1.0083) signifiant que le conditionnement global de ces plans est équivalent mais que la suite de Sobol présente des points plus proches. Le plan WSP qui présentait les meilleurs critères avant le repliage conserve cette qualité après repliage avec des valeurs *Mindist* ($= 1.4307$) et *MoyMin* ($= 1.6237$) élevées et bien supérieures aux valeurs obtenues pour les autres plans et se rapprochant des valeurs du plan de référence construit en 22D avec 60 points ($Mindist = 1.644$; $MoyMin = 1.6709$).

Si nous diminuons le nombre de facteurs conservés $30D \rightarrow 15D$, les plans peuvent être classés comme précédemment avec la suite de Faure qui présente les plus mauvais critères ($Mindist = 0.1249$; $MoyMin = 0.2091$). La matrice aléatoire ($Mindist = 0.6355$; $MoyMin = 1.0142$), la suite de Sobol ($Mindist = 0.7858$; $MoyMin = 1.0317$) et le plan rLHS ($Mindist = 0.7550$; $MoyMin = 1.0252$) présentent des critères proches mais pour ce repliage la matrice aléatoire présente une distance minimale entre points inférieure à celles de la suite de Sobol et du plan rLHS.

Si nous considérons un repliage de $30D \longrightarrow 7D$, nous obtenons un classement des plans différent de celui observé pour les autres pourcentages de réduction. La suite de Faure présente des valeurs *Mindist* et *MoyMin* égales et faibles ($= 0.0853$) avec une valeur *Coverage* proche de zéro. Les autres plans sont regroupés dans une autre partie des graphes avec des valeurs *Mindist* proches de 0.28, une valeur *MoyMin* aux alentours de 0.5 pour une valeur *Coverage* proche de 0.25.

L'analyse des critères intrinsèques nous permet d'observer que les plans 30D repliés suivent le même classement que les plans initiaux lorsque 22 ou 15 facteurs sont conservés alors qu'ils forment deux groupes pour le repliage le plus sévère avec la suite de Faure qui présente des alignements de points très proches et les autres plans qui sont de qualité équivalente.

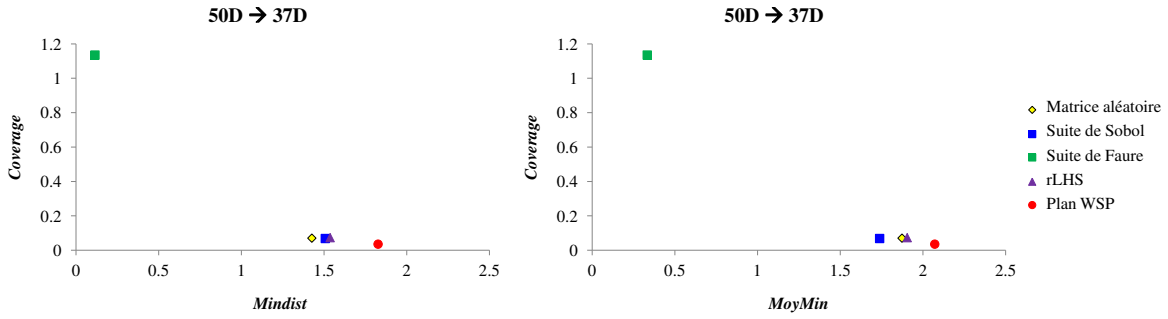
3.3.4.3 Caractérisation des plans 50D repliés

Par le repliage des plans en 50D, nous souhaitons vérifier si les conclusions tirées en 30D demeurent identiques en dimension supérieure. Le tableau 2.19 regroupe les valeurs des critères des plans en 50D repliés dans des sous-espaces en 37D, 25D ou 12D.

Tableau 2.19 – Repliage des plans en **50D** dans des sous-espaces en **37D**, **25D** et **12D**.

Plans initiaux : 50D – 100 points		***Mindist***	***MoyMin***	***Coverage***	R_{min}	R_{moy}
Repliage **37D**	Matrice aléatoire	1.4261	1.8733	0.0704	0.75	0.95
	Suite de Sobol	1.5075	1.7381	0.0683	0.75	0.97
	Suite de Faure	0.1148	0.3330	1.1351	0.11	0.45
	rLHS	1.5366	1.9047	0.0734	0.70	0.91
	Plan WSP	1.8265	2.0714	0.0355	0.74	0.96
Repliage **25D**	Matrice aléatoire	1.1010	1.4437	0.0724	0.75	0.97
	Suite de Sobol	1.1573	1.3780	0.0721	0.73	0.91
	Suite de Faure	0.0943	0.1742	1.4578	0.11	0.34
	rLHS	1.2877	1.4659	0.0667	0.77	0.92
	Plan WSP	1.5210	1.6759	0.0507	0.78	0.94
Repliage **12D**	Matrice aléatoire	0.3894	0.7819	0.1632	0.47	0.89
	Suite de Sobol	0.6495	0.8425	0.1356	0.61	0.87
	Suite de Faure	0.0654	0.0743	1.1954	0.14	0.23
	rLHS	0.5932	0.8166	0.1327	0.62	0.9
	Plan WSP	0.6706	0.9510	0.1446	0.62	0.88

Nous proposons de représenter les valeurs ci-dessus par les représentions graphiques du critère *Coverage* en fonction du critère *Mindist* ou *MoyMin* (figure 2.59).



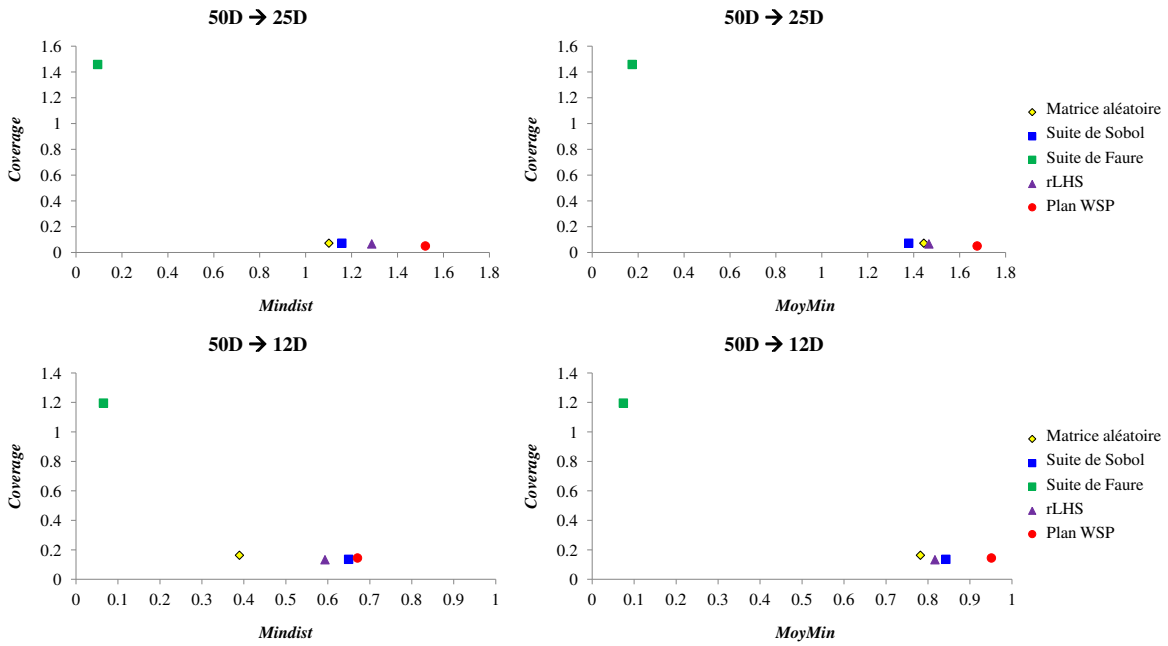


FIGURE 2.59 – Évolution des critères intrinsèques de qualité des plans 50D repliés.

Si nous considérons le repliage des plans en 50D (tableau 2.19 et figure 2.59), les observations ressemblent à celles des plans 30D repliés en 22D et 15D et principalement pour les suites de Faure qui présentent systématiquement les plus mauvais critères avec de faibles valeurs *Mindist* et *MoyMin* et des valeurs *Coverage* élevées (supérieures à 1). Un groupe constitué par la matrice aléatoire, la suite de Sobol et le plan rLHS présente de meilleurs critères, néanmoins moins bons que ceux des plans WSP. Si nous étudions le repliage sur la moitié des facteurs, la suite de Faure présente une valeur *Mindist* égale à 0.0943 qui est bien plus faible que celles obtenues pour la matrice aléatoire (= 1.1010), la suite de Sobol (= 1.1573) et le plan rLHS (= 1.2877). Cette grande différence des valeurs *Mindist* pour un même nombre de points dans le même espace des variables nous mène à penser que la suite de Faure présente de nombreux amas ou des alignements de points très proches. Cette remarque est confirmée par la valeur *MoyMin* qui est égale à 0.1742 pour la suite de Faure alors qu'elle est de l'ordre de 1.4 pour la matrice aléatoire, la suite de Sobol et le plan rLHS. Le plan WSP a des valeurs *Mindist* et *MoyMin* respectivement égales à 1.5210 et 1.6759 qui se rapprochent des valeurs du plan de référence construit en 25D avec 100 points ($Mindist = 1.7184$; $MoyMin = 1.7455$).

La comparaison de la qualité intrinsèque des plans en 50D repliés nous permet de constater que les plans conservent le même classement et ce quel que soit le pourcentage de facteurs conservés.

3.3.5 Étude des sous-espaces en dimension *D'*

Les deux dernières colonnes des tableaux 2.17 à 2.19 contiennent les valeurs des ratios R_{min} et R_{moy} calculés à partir de la projection ACC des plans repliés. Nous rappelons que le ratio R_{min} compare le minimum des distances minimales entre deux points, soit le critère *Mindist*, du plan obtenu par projection ACC et celui du plan de référence. Une valeur proche de 1 signifie que la plus petite distance entre 2 points est comparable à celle du plan de référence alors qu'une faible valeur indique la présence de points très proches et donc d'amas. Nous complétons l'interprétation de ce ratio par le calcul du R_{moy} qui

considère le critère *MoyMin*. Une valeur R_{moy} équivalente à celle du R_{min} indiquera une homogénéité de la répartition des distances minimales et renseigne sur le conditionnement global. Par ailleurs, si les deux ratios sont proches de 1, la distribution de points sera proche du plan de référence et donc de l'uniformité.

La figure 2.60 permet de comparer les valeurs des ratios R_{min} et R_{moy} calculés à partir de la projection ACC des plans repliés, pour les différents plans étudiés et les différents degrés de réduction de dimension.

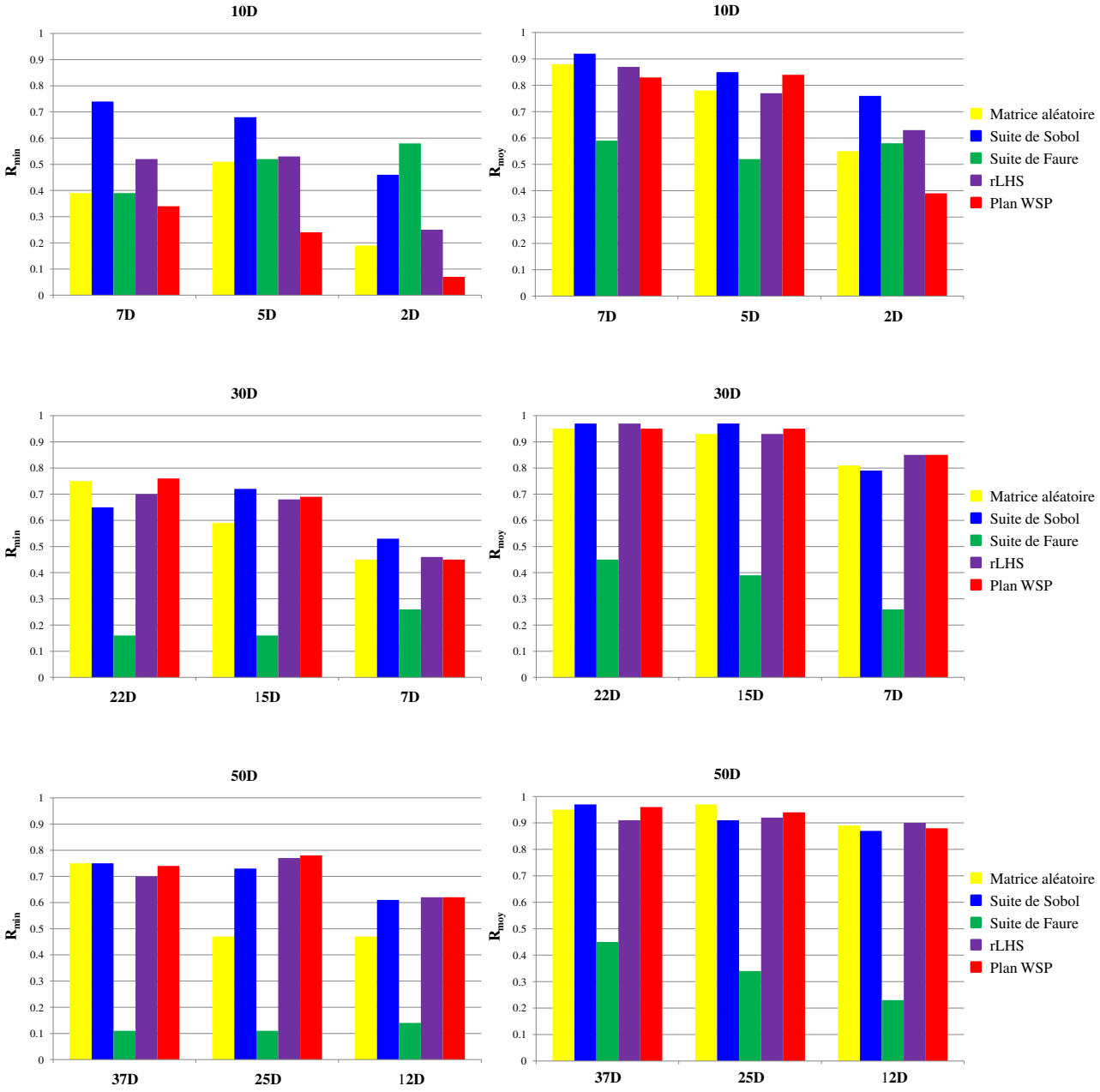


FIGURE 2.60 – Comparaison des ratios R_{min} et R_{moy} calculés pour les plans repliés en 10D, 30D et 50D repliés sur une majorité (environ 75%), la moitié et peu de facteurs (environ 25%).

D'une manière générale, les valeurs R_{min} sont inférieures aux valeurs R_{moy} et nous observons que le comportement des plans en 10D diffère un peu de celui des plans en 30 et 50D. D'autre part, on peut voir que les différents plans initiaux ne se comportent pas de manière identique et dans certains cas, le degré de réduction de dimension a son influence aussi. Ceci nous amène à nous interroger sur

l'influence de la dimension des plans initiaux, de la nature du plan et du nombre D' de facteurs conservés dans le nouvel espace. Pour répondre à ces questions et faciliter l'interprétation de la figure 2.60, nous proposons dans un premier temps d'étudier l'impact de la dimension D des plans, puis l'influence du nombre de facteurs constituant le nouvel espace D' et la nature du plan.

3.3.5.1 Étude de la dimension D et de la dimension D'

Nous avons choisi d'étudier l'impact de la dimension D de l'espace initial et du degré de réduction de dimension pour deux types de plans : les plans issus de l'algorithme WSP qui présentent de bons critères d'uniformité quelle que soit la dimension et les suites de Faure, qui présentent les plus mauvais critères initiaux. Les figures 2.61 et 2.62 permettent de comparer l'évolution des ratios R_{min} et R_{moy} pour les deux types de plans en fonction du pourcentage de facteurs conservés. Comme nous l'avons montré précédemment, la comparaison des valeurs R_{min} et R_{moy} nous renseignera sur le conditionnement local et global des plans résultant des différents repliages.

– **Plans WSP :**

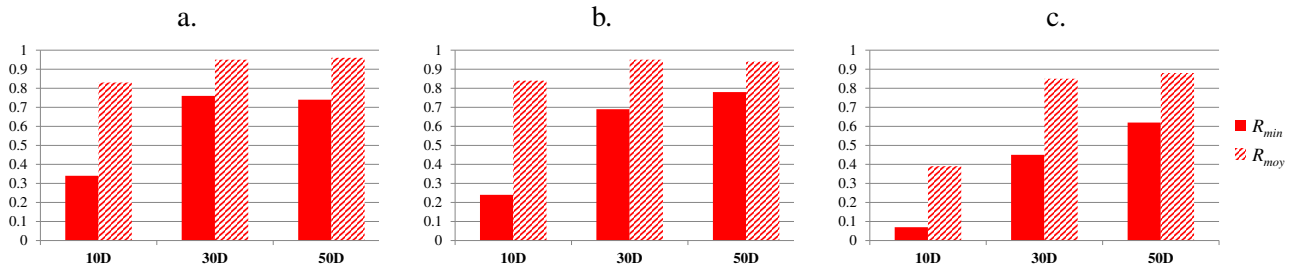


FIGURE 2.61 – Évolution des ratios R_{min} et R_{moy} calculés à partir des plans WSP repliés **a)** sur un grand nombre des facteurs, **b)** sur la moitié des facteurs, **c)** sur peu de facteurs.

Ces diagrammes montrent que le comportement en 10D est différent de celui observé en 30 et 50D. En effet, on constate qu'en 10D la qualité des plans repliés, quelle que soit la proportion de facteurs conservés, est toujours moins bonne (R_{min} plus faible) que celle des plans en 30 et 50D, qui eux se révèlent équivalents. D'autre part, on observe qu'une forte réduction du nombre de facteurs (> 50%) entraîne une dégradation importante de la structure, avec la création d'amas (faible valeur de R_{min}) et un conditionnement global qui s'éloigne de l'uniformité (R_{moy} faible).

– **Suite de Faure :**

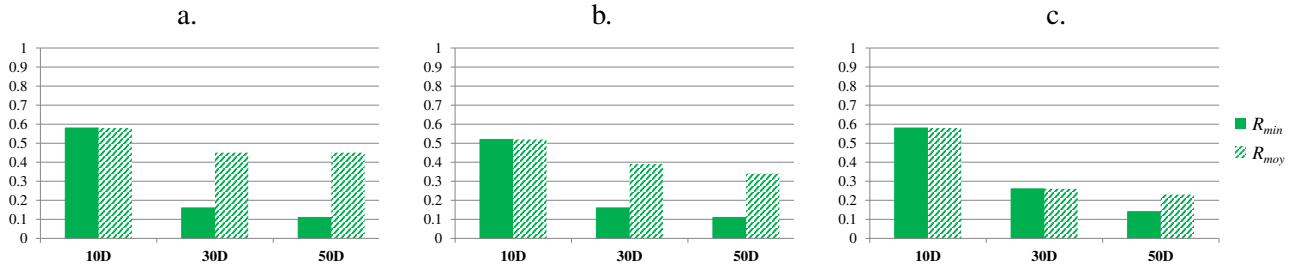


FIGURE 2.62 – Évolution des ratios R_{min} et R_{moy} calculés à partir des suites de Faure repliées **a)** sur une majorité des facteurs, **b)** sur la moitié des facteurs, **c)** sur peu de facteurs.

Pour des conditions identiques de repliage, nous constatons (figure 2.62) que les valeurs R_{min} et R_{moy} sont beaucoup plus faibles que celles obtenues pour les plans WSP. Le repliage des suites de Faure montre deux types de comportement : le premier, observable en 10D avec $R_{min} = R_{moy}$, alors que pour les autres dimensions, nous obtenons des valeurs R_{min} qui sont inférieures aux valeurs R_{moy}.

L'étude de ces ratios montre que le repliage des suites de Faure conduit à des plans globalement très mal conditionnés caractérisés par les faibles valeurs des ratios R_{moy} et présentant de nombreux amas (R_{min} faible).

3.3.5.2 Étude de la nature du plan

Pour étudier l'effet de la nature du plan sur les qualités des plans projetés, nous fixons la dimension des plans initiaux à 30 et nous considérons un repliage en conservant la moitié des facteurs. La figure 2.63 représente les valeurs des ratios pour les différents plans étudiés.

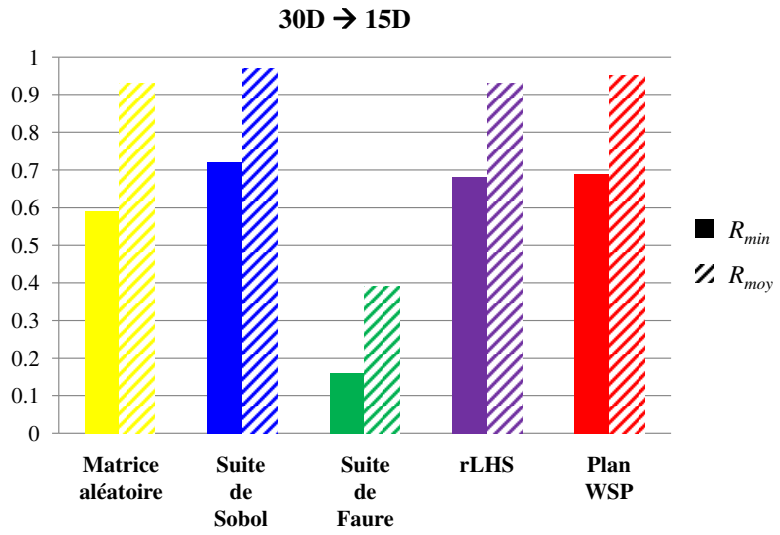


FIGURE 2.63 – Évolution des ratios R_{min} et R_{moy} calculés pour le repliage des plans de 30D vers un sous-espace en 15D.

Si nous comparons tous les plans, nous observons (figure 2.63) que les ratios les plus faibles sont toujours obtenus pour la suite de Faure.

3.3.5.3 Synthèse

D'une manière générale, lorsque la dimension des plans initiaux augmente ($D \geq 30$) les conséquences du repliage, à savoir la création d'amas et/ou de lacunes, sont réduites. Le comportement des ratios R devient alors similaire pour un type de plan donné : les valeurs les plus élevées sont obtenues pour les suites de Sobol, les hypercubes latins et les plans WSP alors que les suites de Faure présentent systématiquement de très faibles ratios, synonymes d'une mauvaise répartition des points. Toutefois, effectuer un repliage en conservant plus de la moitié des facteurs ne semble pas avoir de répercussion en termes de répartition uniforme des points alors qu'un repliage en ne conservant qu'un très faible nombre de facteurs va induire une perte d'uniformité, due à une concentration des points dans l'espace.

3.3.6 Perspectives pour la réparation des plans en dimension *D'*

L'étude des sous-espaces en D' dimensions nous a permis de montrer que le repliage peut engendrer la création d'amas de points ou de lacunes. Si nous souhaitons conserver une répartition uniforme des points, nous devons supprimer les amas et combler les lacunes. Dans le "catalogue" de méthodes présentées dans la partie 1, nous disposons d'une méthode pour éliminer les amas et d'une autre pour remplir les zones lacunaires, mais nous nous demandons dans quel ordre ces méthodes doivent être appliquées. Nous proposons alors deux stratégies :

- la **première stratégie** débute par l'élimination des amas pour ensuite combler les zones lacunaires,
- la **deuxième stratégie** consiste à remplir les lacunes puis, à supprimer les amas.

Nous avons déjà évoqué les avantages et inconvénients de ces deux stratégies mais dans le cadre de cette étude de repliage nous souhaitons effectuer une étude fondamentale et exhaustive. Pour cela, nous choisissons d'appliquer ces stratégies sur les plans en 50 dimensions et 100 points que nous avons préalablement repliés vers un sous-espace en 37D, 25D et 12D. Pour chaque plan, nous comparerons le nombre de points restants après chaque étape puis avec les plans résultant des étapes de réparation, nous calculerons les ratios R_{min} et R_{moy}, à partir de la projection en Composantes Curvilignes et d'une matrice de référence avec le même nombre de points.

3.3.6.1 Première stratégie de réparation

Nous débuterons par le repliage le moins sévère qui conserve une majorité de facteurs.

Tableau 2.20 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **37D** selon la **stratégie 1**.

Stratégie 1 : Réparation des plans en 50D après repliage sur 37 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après élimination des amas	Après remplissage des lacunes					
Matrice aléatoire	**45**	**509**	0.032	0.040	0.040	**0.8**	**1**
Suite de Sobol	**1**						
Suite de Faure	**3**	**106**	0.071	0.087	0.091	**0.78**	**0.96**
rLHS	**54**	**546**	0.031	0.038	0.038	**0.82**	**0.99**
Plan WSP	**72**	**167**	0.057	0.069	0.070	**0.82**	**0.99**

La première étape de réparation conduit à des plans sans amas avec des nombres de points toujours beaucoup plus faibles et différents selon la nature du plan. La matrice aléatoire, le plan rLHS et le plan WSP conservent le plus de points avec respectivement 45, 54 et 72 points alors qu'il ne reste qu'un seul point pour la suite de Sobol et 3 points pour la suite de Faure. Cette diminution du nombre de points signifie que tous les plans après repliage présentent des amas, qui sont plus ou moins nombreux selon le plan considéré. Pour les suites à faible discrépance, cette première étape d'élimination des amas ne conserve que très peu de points ce qui peut s'expliquer par deux phénomènes. Pour cela, nous proposons

tout d'abord de représenter sur la figure 2.64 la projection ACC de la suite de Faure repliée sur 37 facteurs.

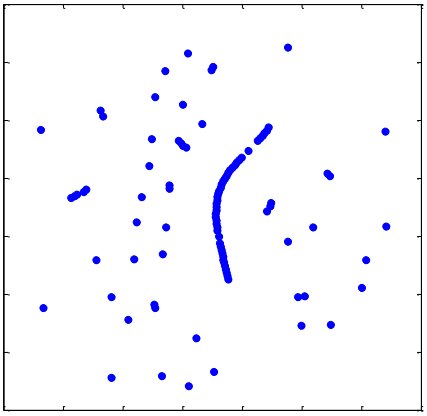


FIGURE 2.64 – Projection par ACC de la suite de Faure en 37D et 100 points.

Cette projection (figure 2.64) montre que les 100 points constituant la suite de Faure repliée sont mal répartis avec de nombreux amas, des alignements de points et des zones lacunaires. Ainsi, de nombreux points sont extrêmement proches et sont alors supprimés par l'algorithme de sélection WSP lors de l'étape de suppression d'amas. Pour la suite de Sobol repliée sur 37 facteurs, nous avions obtenu des valeurs $R_{min} = 0.75$ et un $R_{moy} = 0.97$ (tableau 2.19) qui traduisent un bon conditionnement des points or après cette première étape nous observons que seul le point de départ de l'algorithme WSP est sélectionné ce qui signifie que les 99 points éliminés se situaient à une distance inférieure à la valeur d_{min} de référence de ce point. Dans toute cette étude, nous avons choisi de toujours fixer le point au centre du domaine ou son plus proche voisin comme point de départ de l'algorithme, mais le choix d'un autre point de départ pourrait conduire à des résultats différents. Sur la figure 2.65, nous proposons de comparer les points sélectionnés par l'algorithme WSP pour une même distribution de points et une même valeur d_{min} lorsque nous changeons le point initial.

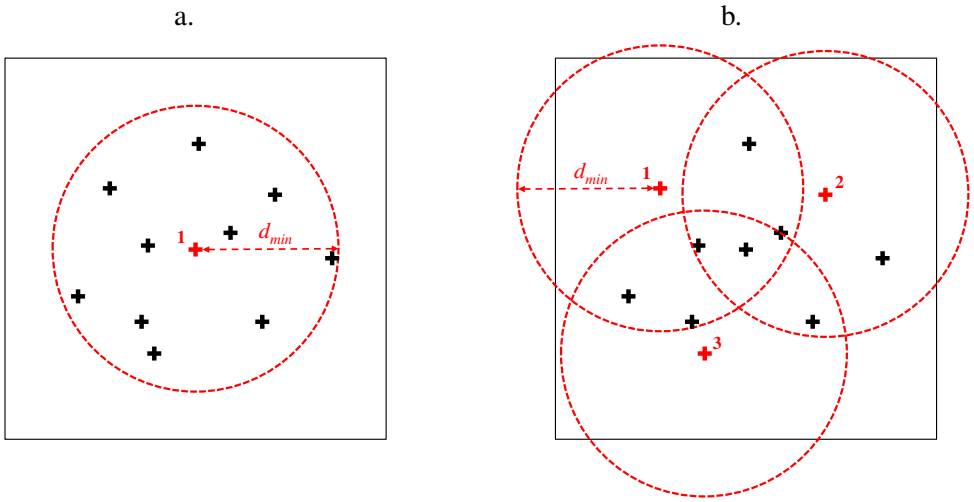


FIGURE 2.65 – Comparaison du choix du point de départ de l'algorithme WSP pour l'étape de suppression des amas pour une même distribution de points. **a)** Le point au centre est choisi comme point initial de l'algorithme WSP et seul ce point est sélectionné. **b)** Le point initial n'est plus le point au centre ce qui conduit à la sélection de 3 points. Les points sélectionnés sont représentés en rouge.

Sur la figure ci-dessus, nous constatons que le départ du point au centre ou son plus proche voisin (figure 2.65 a.) permet de ne sélectionner que ce point alors que le départ d'un autre point situé plus à la périphérie de cet ensemble de points permet d'en sélectionner trois (points rouges).

Dans cette étude, nous avons choisi de toujours fixer comme point de départ de l'algorithme le point au centre ou son plus proche voisin car en parallèle nous construisons des plans de référence selon cette démarche ce qui nous permet de comparer des ensembles de points obtenus selon la même méthode.

Pour la deuxième étape de réparation qui consiste à combler les zones lacunaires, nous observons que de nombreux points sont ajoutés à la matrice aléatoire et au plan rLHS conduisant à des plans réparés qui comptent respectivement 464 et 492 points, ce qui signifie que le repliage de ces plans engendre la création de nombreuses lacunes qui peuvent aussi être créées lors de la première étape. Pour la suite de Faure et le plan WSP, cette deuxième étape ajoute une centaine de points ramenant ces plans à 106 et 167 points. Pour la suite de Sobol, nous n'avons pas comblé les zones lacunaires car à partir d'un seul point, cette étape serait synonyme d'une reconstruction totale du plan. La présence de zones lacunaires peut être la conséquence du repliage et dans ce cas les lacunes étaient présentes avant la réparation du plan, ou elles peuvent être créées lors de l'étape d'élimination des amas. En effet, la suppression de certains points, peut conduire à une augmentation de la valeur maximale des distances minimales.

A partir des plans totalement réparés selon cette stratégie, nous effectuons une ACC pour comparer les valeurs des ratios avant et après réparation. D'une manière générale, pour la matrice aléatoire, le plan rLHS et le plan WSP, les R_{min} augmentent de (0.75 à 0.8) et les R_{moy} qui étaient déjà supérieurs à 0.9 se rapprochent de 1. Pour le suite de Faure ces ratios étaient les plus faibles avant réparation avec $R_{min} = 0.11$ et $R_{moy} = 0.46$, qui traduisaient la présence à la fois d'amas et d'un mauvais conditionnement mais deviennent équivalents aux ratios des autres plans. Néanmoins cette nette amélioration doit être interprétée avec précaution car dans ce cas la stratégie de réparation se rapproche d'une reconstruction de plan.

Nous proposons de vérifier si les plans se comportent de la même manière lorsque nous effectuons un repliage plus sévère, à savoir ne conserver que la moitié des facteurs ($50D \rightarrow 25D$).

Tableau 2.21 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **25D** selon la **stratégie 1**.

Stratégie 1 : Réparation des plans en 50D après repliage sur 25 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après élimination des amas	Après remplissage des lacunes					
Matrice aléatoire	**23**	**186**	0.054	0.065	0.067	**0.81**	**0.97**
Suite de Sobol	**1**						
Suite de Faure	**3**	**104**	0.075	0.087	0.091	**0.83**	**0.95**
rLHS	**30**	**199**	0.053	0.063	0.064	**0.83**	**0.98**
Plan WSP	**41**	**129**	0.065	0.079	0.081	**0.81**	**0.97**

Tout comme le repliage vers une sous-espace en 37D, nous observons que la première étape d'élimination des amas, élimine quasiment tous les points pour les suites à faible discrépance en ne conservant que le point initial de l'algorithme pour la suite de Sobol et 3 points pour la suite de Faure alors que pour les autres plans, davantage de points sont conservés avec 23 points pour la matrice aléatoire, 30 pour le plan rLHS et 41 pour le plan WSP. Si nous comparons le nombre de points restants après cette première étape pour un repliage sur une majorité de facteurs ($50D \rightarrow 37D$) ou sur la moitié ($50D \rightarrow 25D$), nous constatons qu'il reste moins de points dans le sous-espace en 25D ce qui signifie qu'un repliage plus sévère favorise la création d'amas.

La deuxième étape de remplissage des lacunes ajoute des points pour tous les plans (sauf la suite de Sobol pour laquelle nous n'avons pas effectué cette étape) avec l'ajout respectivement de 163, 101, 169 et 88 points pour la matrice aléatoire, la suite de Faure, le plan rLHS et le plan WSP. Toutefois, le nombre de points ajouté est bien plus faible que pour les plans en 37D ce qui peut signifier que le repliage plus sévère minimise les zones de vide.

Si nous projetons ces nouvelles distributions de points par ACC, nous constatons que tous les plans présentent des ratios élevés, synonyme d'une distribution proche du plan de référence avec le même nombre de points.

La dernière application de cette stratégie est effectuée sur le repliage le plus sévère qui ne conserve que 12 facteurs.

Tableau 2.22 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **12D** selon la **stratégie 1**.

Stratégie 1 : Réparation des plans en 50D après repliage sur 12 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après élimination des amas	Après remplissage des lacunes					
Matrice aléatoire	**14**	**79**	0.087	0.101	0.108	**0.80**	**0.94**
Suite de Sobol	**10**	**98**	0.078	0.089	0.092	**0.85**	**0.96**
Suite de Faure	**5**	**101**	0.076	0.087	0.094	**0.81**	**0.93**
rLHS	**20**	**73**	0.097	0.107	0.112	**0.86**	**0.96**
Plan WSP	**37**	**108**	0.076	0.088	0.089	**0.86**	**0.99**

A partir des résultats regroupés dans le figure 2.22, nous constatons que beaucoup moins de points sont conservés après l'étape d'élimination des amas que pour le repliage des plans sur 37 et 25 facteurs, ce qui confirme ce que nous avions déjà constaté à savoir qu'un repliage plus sévère favorise la création d'amas de points. Par ailleurs, la deuxième étape conduit à des plans réparés avec respectivement 79 et 73 points pour la matrice aléatoire et le plan rLHS qui présentent donc moins de lacunes que la suite de Sobol, la suite de Faure et le plan WSP qui comptent 98, 101 et 108 points.

Le calcul des ratios après le projection ACC des plans totalement réparés selon cette première stratégie, conduit à des valeurs qui sont très proches pour tous les plans.

Avec cette première stratégie, nous constatons qu'il est difficile de réparer tous les plans de la même manière puisque nous ne pouvons pas obtenir des ensembles comparables en termes de nombre de points. Toutefois, nous avons pu mettre en évidence que le repliage sur très peu de facteurs ($50D \rightarrow 12D$) crée plus d'amas mais moins de lacunes que pour un repliage en conservant une majorité de facteurs ($50D \rightarrow 37D$).

3.3.6.2 Deuxième stratégie de réparation

La première stratégie nous a permis de mettre en exergue les conséquences du repliage d'un plan, par la création d'amas et de lacunes. Nous savons que les lacunes peuvent être soit présentes dans le plan replié, soit résulter de la suppression des amas. Afin de comparer l'impact du repliage sur la présence des lacunes, nous proposons la deuxième stratégie qui consiste à combler les zones de vide pour ensuite supprimer les points trop proches. Nous pourrons alors savoir quels sont les plans les plus sensibles au repliage selon le pourcentage de facteurs conservés.

Nous débutons par le repliage le moins sévère qui conserve une majorité de facteurs ($50D \longrightarrow 37D$).

Tableau 2.23 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **37D** selon la **stratégie** 2.

Stratégie 2 : Réparation des plans en 50D après repliage sur 37 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après remplissage des lacunes	Après élimination des amas					
Matrice aléatoire	**483**	**424**	0.035	0.043	0.043	**0.82**	**1**
Suite de Sobol	**160**	**61**	0.102	0.122	0.123	**0.83**	**0.99**
Suite de Faure	**159**	**62**	0.093	0.114	0.123	**0.75**	**0.93**
rLHS	**509**	**462**	0.035	0.041	0.041	**0.85**	**1**
Plan WSP	**191**	**163**	0.057	0.070	0.071	**0.81**	**0.98**

A partir du figure 2.23, nous observons clairement que la différence du nombre de points après la première étape implique que les plans repliés ne présentent pas tous le même nombre de lacunes. En effet, environ 60 points sont ajoutés pour les suites à faible discrépance, 91 pour le plan WSP et environ 400 pour la matrice aléatoire et le plan rLHS. Ces premières valeurs, montrent que les suites à faible discrépance sont celles qui présentent le moins de lacunes contrairement à la matrice aléatoire et au plan rLHS. Ces résultats vont dans le même sens que ce que nous avions montré par la première stratégie pour le même repliage.

La deuxième étape d'élimination des amas permet de mettre en évidence que les suites à faible discrépance sont celles qui contiennent le plus d'amas de points ce qu suppression de 100 points environ, alors que 59, 47 et 28 points sont respectivement éliminés pour la matrice aléatoire, le plan rLHS et le plan WSP.

La projection ACC de ces plans réparés, permet de calculer les ratios et nous observons que toutes les valeurs sont très proches.

Tableau 2.24 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **25D** selon la **stratégie 2**.

Stratégie 2 : Réparation des plans en 50D après repliage sur 25 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après remplissage des lacunes	Après élimination des amas					
Matrice aléatoire	**159**	**84**	0.086	0.099	0.101	**0.80**	**0.90**
Suite de Sobol	**122**	**23**	0.172	0.194	0.221	**0.78**	**0.88**
Suite de Faure	**151**	**54**	0.109	0.126	0.13	**0.84**	**0.97**
rLHS	**152**	**85**	0.085	0.097	0.101	**0.84**	**0.96**
Plan WSP	**167**	**108**	0.072	0.086	0.089	**0.81**	**0.97**

A partir des plans en 25D, nous observons clairement que le repliage en conservant la moitié des facteurs ajoute moins de points que pour les plans en 37D ce qui signifie qu'un repliage plus sévère crée moins de lacunes avec 22 points ajoutés pour la suite de Sobol et environ 50 points pour les autres plans.

A partir de la matrice aléatoire, la suite de Faure et le plan rLHS sans lacunes qui comptent quasiment le même nombre de points, la suppression des amas retient respectivement 84, 54 et 85 points ce qui signifie que la suite de Faure présente plus d'amas de points que la matrice aléatoire et le plan rLHS. Pour la suite de Sobol, cette étape supprime 99 points ce qui caractérise toujours le phénomène que nous avons pu mettre en évidence lors de la première stratégie avec une concentration de points dans l'hypersphère de rayon d_{min} de référence autour du point au centre. Pour le plan WSP, 59 points sont éliminés tout comme la première étape de la stratégie 1.

Tableau 2.25 – Réparation des plans en **50D** et 100 points repliés vers un sous-espace en **12D** selon la **stratégie 2**.

Stratégie 2 : Réparation des plans en 50D après repliage sur 12 facteurs	Nombre de points		*Mindist* après ACC et réparation totale	*MoyMin* après ACC et réparation totale	*Mindist* et *MoyMin* de référence pour le nombre de points après réparation totale	Ratio R_{min} après réparation totale	Ratio R_{moy} après réparation totale
	Après remplissage des lacunes	Après élimination des amas					
Matrice aléatoire	**104**	**18**	0.199	0.224	0.249	**0.80**	**0.90**
Suite de Sobol	**101**	**11**	0.290	0.331	0.339	**0.86**	**0.98**
Suite de Faure	**161**	**66**	0.096	0.111	0.115	**0.84**	**0.96**
rLHS	**102**	**23**	0.186	0.214	0.221	**0.84**	**0.97**
Plan WSP	**116**	**53**	0.107	0.123	0.134	**0.80**	**0.91**

Le remplissage des zones de vide à partir des plans en 50D repliés vers un sous-espace en 12D, ne requiert l'ajout de moins de points que pour les pourcentages de facteurs repliés envisagés précédemment. Seule la suite de Faure conduit à l'ajout de 61 points contre 4, 1, 2 et 16 pour la matrice aléatoire, la suite de Sobol, le plan rLHS et le plan WSP. cette observation confirme que le repliage

sur peu de facteurs minimise le nombre de lacunes mais augmente la quantité d'amas de points qui se caractérise par le faible nombre de points restants à l'issue de la seconde étape de réparation.

Après la réparation totale de ces plans, les ratios calculés à partir de la projection ACC présentent des valeurs élevées.

Cette seconde stratégie qui débute par le remplissage des zones de vide permet de s'affranchir de la difficulté que nous avions rencontré pour les suites de Sobol pour lesquelles les points peuvent être regroupés autour du point au centre et ainsi laisser apparaitre des zones déficientes en informations dans le reste du domaine. Toutefois, les observations faites à l'issue de la réparation selon la stratégie 2 permettent de confirmer qu'un repliage sévère ($50D \rightarrow 12D$) crée moins de lacunes qu'un repliage dans un sous-espace de dimension supérieure ($50D \rightarrow 37D$) mais s'accompagne de nombreux amas.

3.3.7 Conclusion

Dans cette étude, nous avons choisi de nous intéresser à des SFD classiques et des plans aléatoires en 10, 30 et 50D à partir desquels nous avons effectué un repliage ($D \longrightarrow D'$). Cette projection s'accompagne d'une modification de la structure du plan et peut conduire à la création d'amas de points et de lacunes. Nous avons alors montré que la qualité intrinsèque des plans repliés dépend à la fois de la nature du plan, de la dimension D mais aussi du pourcentage de facteurs conservés (D').

Pour pallier le mauvais conditionnement des plans en D' dimensions nous avons proposé d'utiliser des méthodes du "catalogue" et notamment l'algorithme WSP afin de supprimer les amas et de combler les lacunes. Ces méthodes de réparation nous ont conduit à envisager deux stratégies qui se différencient par leur ordre d'exécution .

Cette étude nous a permis de mettre en évidence qu'un repliage sur peu de facteurs crée de nombreux amas mais peu de lacunes contrairement à un repliage qui conserve un grand nombre de facteurs. D'autre part, deux stratégies de réparation ont mis en évidence que les conséquences du repliage diffèrent selon la nature du plan initial. En effet, les plans aléatoires et rLHS sont ceux qui présentent le plus de lacunes, alors que la suite de Faure est constituée par de nombreux alignements et de points extrêmement proches.

Les deux stratégies de réparation permettent d'améliorer la répartition des points au sens de l'uniformité, mais le choix de la stratégie dépendra des contraintes expérimentales qui peuvent par exemple limiter le nombre de points à ajouter. En effet, il est inenvisageable de compléter un plan d'expériences à N points par un ajout d'autres N points.

Ces premiers résultats nous permettent de conclure que les SFD classiques ne sont pas systématiquement adaptés au repliage puisqu'en réduisant la dimension nous cassons la structure des plans, pouvant ainsi créer de nombreux amas et lacunes dans les espaces de projection. Aussi, il serait souhaitable de construire de nouveaux plans qui seraient optimaux en D dimensions mais aussi dans toutes les D' dimensions, en garantissant une répartition uniforme des points quels que soient les espaces.

Conclusion et perspectives

Conclusion

Les travaux de recherche présentés dans ce manuscrit portent sur l'étude de données en grande dimension qui se rencontrent dans de nombreux domaines d'application. Nous avons montré qu'en grande dimension les difficultés proviennent principalement du manque d'outils adaptés à ces espaces multidimensionnels, et plus particulièrement de l'insuffisance des méthodes de visualisation et de sélection.

Aussi, nous avons souhaité aborder ces sujets et proposer un catalogue de méthodes et outils qui répondent aux besoins liés à la grande dimension et plus précisément à :

- l'évaluation de la qualité intrinsèque d'un ensemble de points constituant une base de données,
- la visualisation de données multidimensionnelles,
- la sélection de sous-ensembles représentatifs d'un ensemble initial.

Pour évaluer et décrire graphiquement un jeu de données, nous avons proposé deux types d'outils. Tout d'abord, le calcul de **critères intrinsèques**, *Mindist*, *MoyMin*, *Coverage* ou *Ecart-type*, qui sont basés sur des calculs de distances euclidiennes et qui renseignent sur l'uniformité d'une distribution de points dans un espace multidimensionnel, quelle que soit la dimension. Puis, l'**Analyse en Composantes Curvilignes** comme méthode de visualisation des données par projection dans un espace bidimensionnel, avec les avancées que nous avons proposées pour compléter la simple analyse graphique.

Une autre particularité de la grande dimension peut être la surabondance d'information apportée par les nombreuses expériences (ou simulations) et/ou le nombre important de paramètres (ou variables d'entrée), qui peut nécessiter une sélection judicieuse de sous-ensembles de points (lignes) ou de variables (colonnes). Parmi les différentes méthodes de sélection existantes, nous avons retenu l'algorithme **WSP** qui s'avère performant même en grande dimension pour sélectionner des sous-ensembles de points avec la garantie d'une répartition uniforme dans l'espace des variables et l'algorithme **V-WSP** qui permet de ne retenir que les variables d'entrée les moins corrélées. Ces algorithmes ont été adaptés et améliorés pour répondre aux problématiques de la grande dimension, notamment avec l'algorithme **aWSP** qui permet de densifier plus ou moins certaines zones de l'espace multidimensionnel.

D'autre part, nous nous sommes intéressés à la non-uniformité d'une distribution quelconque de points, qui se caractérise généralement par la présence d'amas et/ou de lacunes. Dans ce contexte, nous avons proposé d'adapter l'algorithme WSP pour éliminer les amas de points et enrichir les zones déficientes en information.

L'application de ces algorithmes sur des cas d'étude réels issus de domaines variés a permis de mettre en évidence les avantages et les inconvénients des différentes méthodes proposées.

Tout d'abord, nous nous sommes intéressés à des données relatives aux études de Relations Structure-Activité (QSAR) pour lesquelles le nombre de descripteurs est souvent très important, avec un ensemble d'individus imposé par le cas d'étude. Ces contraintes nécessitent, soit la visualisation des individus par l'Analyse en Composantes Curvilignes, soit la sélection pertinente d'un sous-ensemble de descripteurs pour la construction d'un modèle prédictif, par l'algorithme V-WSP.

Nous avons ensuite réalisé une étude comparative des méthodes de sélection de points dans le cas de données spectroscopiques, pour la construction d'ensembles de calibration et de validation à des fins de modélisation. Cette étude a montré que les différents algorithmes de sélection conduisent à des modèles de qualité équivalente et ne se différencient que par les temps de calcul.

Enfin, nous avons testé les algorithmes de "réparation" dans le contexte des plans d'expériences uniformes pour la simulation numérique. L'algorithme WSP a permis de réparer des plans non-uniformes par construction ou après une étape de "repliage" suite à une étude de sensibilité.

Perspectives

Les différents cas d'étude ont permis de montrer le potentiel des méthodes proposées dans la première partie de ce manuscrit, mais aussi leurs limites ou leurs insuffisances. Aussi, il serait intéressant de compléter ce catalogue ou d'amender encore les algorithmes pour rendre l'exploitation des données en grande dimension plus aisée et plus fiable.

— Les critères que nous avons choisis de retenir reposent sur le calcul des distances euclidiennes entre les points d'une distribution. Or, dans ce travail nous avons pu mettre en évidence qu'il peut être délicat d'interpréter seulement ces critères. Des outils complémentaires seraient donc utiles pour sonder tout l'espace et avoir ainsi une connaissance en toutes zones du domaine, en considérant par exemple, la densité d'information. Les recherches pourraient porter sur le découpage de l'espace en sous-volumes de taille égale (figure 2.66) pour en évaluer le remplissage en termes de densité locale.

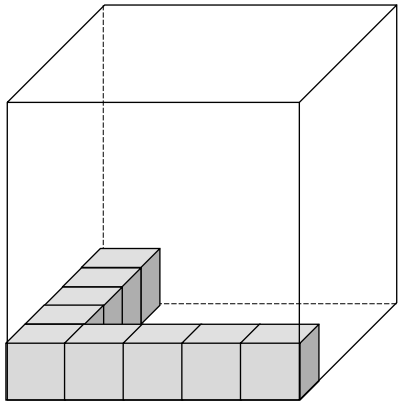


FIGURE 2.66 – Illustration du découpage d'un espace en trois dimensions.

La difficulté de cette méthode reposera sur le choix de la taille des sous-volumes : des volumes trop petits conduiraient à détecter de nombreuses zones vides alors que des volumes trop grands pourraient conduire à une interprétation incertaine.

Nous pourrions aussi envisager d'autres critères qui prennent en compte l'environnement du point étudié comme par exemple la métrique de Mahalanobis localement centrée, proposée par Todeschini et al. [106].

— Un autre point qui nécessite d'être complété porte sur l'algorithme de sélection WSP et le choix judicieux du point de départ et de la distance minimale séparant les points sélectionnés, d_{min}. En effet, nous avons constaté que selon le point de départ de l'algorithme, le sous-ensemble sélectionné diffère et par là-même, ses qualités intrinsèques. Dans nos études, nous avons toujours choisi comme

point de départ, le point central ou son plus proche voisin. Ainsi, dans l'étude du "repliage", nous avons constaté que l'élimination des amas pour les suites de Sobol "repliées" conduit à ne retenir que le point de départ de l'algorithme WSP. En effet, nous montrons que les suites de Sobol "repliées" concentrent tous leurs points autour du point central. Nous nous interrogeons donc sur la bonne démarche à suivre quant à la position du point de départ de l'algorithme WSP lors des étapes de réparation. Pour tenter de répondre à cette question, nous avons réitéré cette étape d'élimination des amas pour la suite de Sobol en 50D "repliée" vers un sous-espace en 25D en envisageant différents points de départ : le point le plus éloigné du centre ou l'un des deux points les plus éloignés dans l'espace ou un point choisi aléatoirement. Nous avons reporté dans le tableau 2.26, la distance entre le point au centre et le point de départ choisi ainsi que le nombre de points restants après l'élimination des amas par l'algorithme.

Tableau 2.26 – Étape d'élimination des amas appliquée sur la suite de Sobol en 50D et 100 points "repliée" sur la moitié des facteurs, en considérant plusieurs points de départ pour l'algorithme WSP.

Point de départ	**Distance par rapport au centre du domaine**	**Nombre de points restants après élimination**
Point au centre	0	1
Point le plus éloigné du centre	1.717	27
Un des deux points les plus éloignés dans l'espace des variables	1.637	28
Un point choisi aléatoirement	1.642	25

Ces résultats montrent que le nombre de points restants après l'étape d'élimination des amas dépend du point de départ de l'algorithme. Il serait donc intéressant de réaliser une étude plus complète en considérant différents points de départ. Comme pour l'algorithme V-WSP, on pourrait obtenir autant de solutions que de points constituant la matrice candidate et le choix du sous-ensemble le plus pertinent pourrait se faire selon des critères intrinsèques et des critères économiques.

Le deuxième paramètre de l'algorithme WSP est la valeur d_{min}, la distance minimale séparant les points sélectionnés. Dans toutes nos applications, nous avons choisi la valeur d_{min} en fonction du critère *Mindist* d'un plan considéré comme référence. Nous pourrions envisager de faire varier la valeur d_{min} pour étudier les conséquences sur le nombre de points sélectionnés. Ainsi, pour comparer les solutions comportant différents nombres de point, nous pourrions calculer leur *écart-type* et suivre son évolution (figure 2.67). Cette information pourrait être utile et tout particulièrement dans le cas où le nombre de points est limité par des contraintes de faisabilité ou économiques. Nous proposons de reconsidérer la matrice en 2D et 100 points que nous avons étudiée dans la Partie 1. Si nous utilisons l'algorithme WSP pour sélectionner des points en faisant varier la valeur d_{min} de 0.05 à 0.20, nous sélectionnons de 74 à 17 points. La figure 2.67 représente l'évolution de l'*écart-type* des différentes solutions en fonction des valeurs d_{min} respectives.

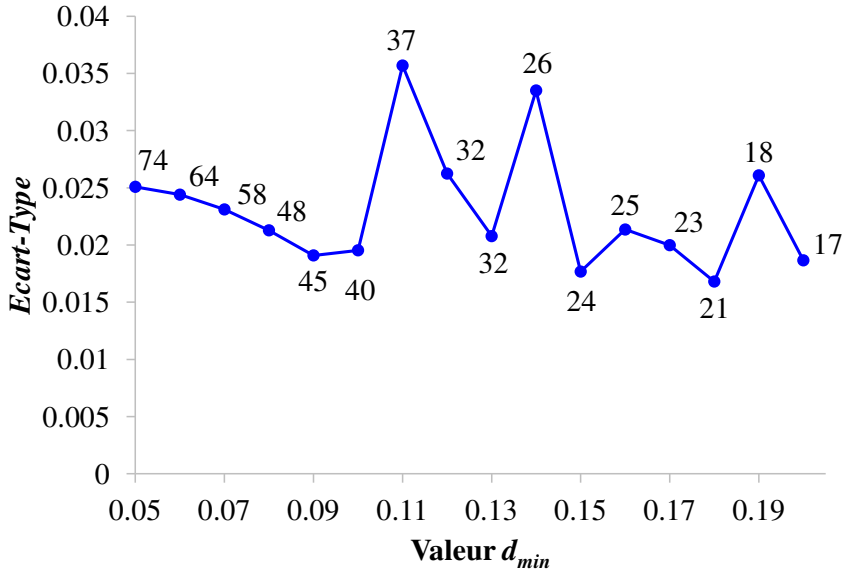


FIGURE 2.67 – Évolution de l'*écart-type* en fonction de la valeur d_{min} choisie pour l'algorithme WSP. Le nombre de points sélectionnés est ajouté sur la courbe.

Nous constatons que l'écart-type présente des variations avec des "cassures" correspondant à de faibles valeurs pour les solutions à 45, 40, 32, 24 et 21 points. Le choix de la "meilleure" solution selon ce critère dépendra alors du nombre d'expériences ou de simulations que l'expérimentateur est prêt à réaliser.

De même, dans le cas du comblement de lacunes, la valeur d_{min} sera imposée par le nombre de points qu'il serait envisageable d'ajouter au plan "à réparer". Comme précédemment, nous pourrions suivre l'évolution du nombre de points à ajouter en fonction de la valeur d_{min} (figure 2.68) et il faudrait alors un critère du type "qualité/prix", pour aider dans le choix du nombre minimum de points à ajouter pour obtenir une structure de qualité acceptable en termes d'uniformité.

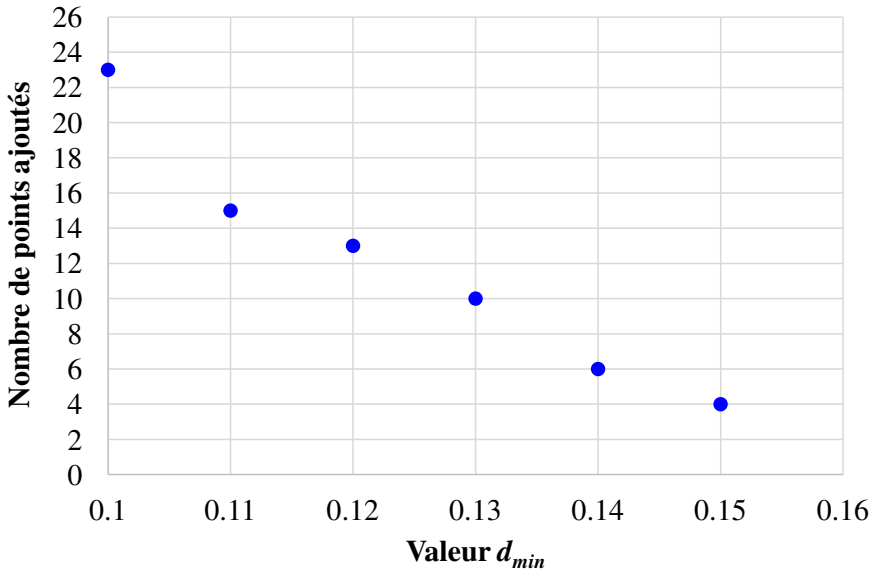


FIGURE 2.68 – Nombre de points ajoutés lors de l'étape du remplissage des lacunes pour différentes valeurs d_{min}.

Au-delà du choix des paramètres d'entrée de l'algorithme WSP (d_{min} et point de départ), nous nous interrogeons aussi sur une autre utilisation de l'algorithme dans le cadre du traitement d'images multispectrales par une approche par blocs. Nous nous posons alors la question de savoir si l'algorithme WSP ne pourrait pas être utilisé et adapté pour répartir N_c points candidats en g groupes de telle manière que chaque groupe soit représentatif de l'ensemble initial. Dans ce cas, nous devons repenser complètement l'algorithme WSP. Cette nouvelle problématique soulève des questionnements notamment sur le principe même de construction des groupes et sur la démarche à suivre : est-il préférable de construire les g groupes en parallèle ou faut-il appliquer un premier algorithme WSP puis à partir des points non sélectionnés appliquer une seconde fois l'algorithme en envisageant un autre point de départ.

— Dans ce travail, nous nous sommes aussi intéressés à l'utilisation des méthodes de sélection pour réaliser un échantillonnage et construire des sous-ensembles de calibration et de validation qui comptent respectivement 80% et 20% des données initiales. La comparaison des performances des modèles PLS ne montre pas de réelles différences entre les méthodes de sélection. Nous nous sommes alors posé la question de savoir si les résultats sont sensibles aux changements des pourcentages de répartition des données. Pour ce faire, nous proposons de reconsidérer la base de données, décrite dans le chapitre 2 de la deuxième partie de ce manuscrit, qui compte 231 spectres infrarouges pour lesquels l'absorbance à 800 nombres d'onde est mesurée. Nous construisons de nouveaux sous-ensembles de calibration et validation en respectant une répartition de 60% et 40% des données initiales selon les mêmes stratégies puis nous avons calculé des modèles de régression PLS. Les valeurs des critères *a posteriori* sont comparées à celles obtenues pour une répartition 80%/20% sur les figures 2.69 à 2.71.

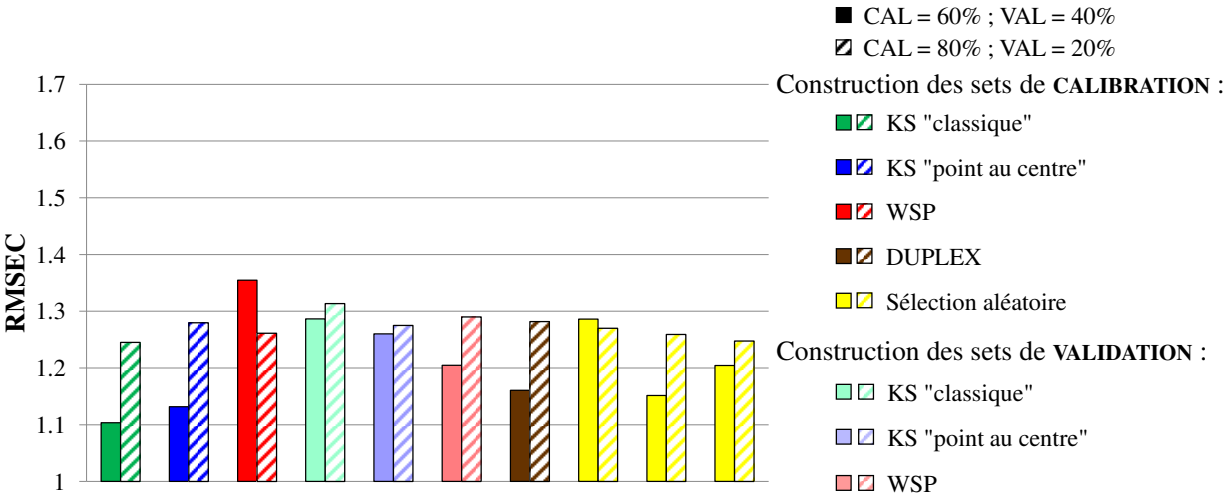


FIGURE 2.69 – RMSEC obtenus à partir des sous-ensembles de calibration pour l'analyse de la réponse "Y1". Les critères représentés sont calculés à partir des sous-ensembles de calibration contenant 60% et 80% des données.

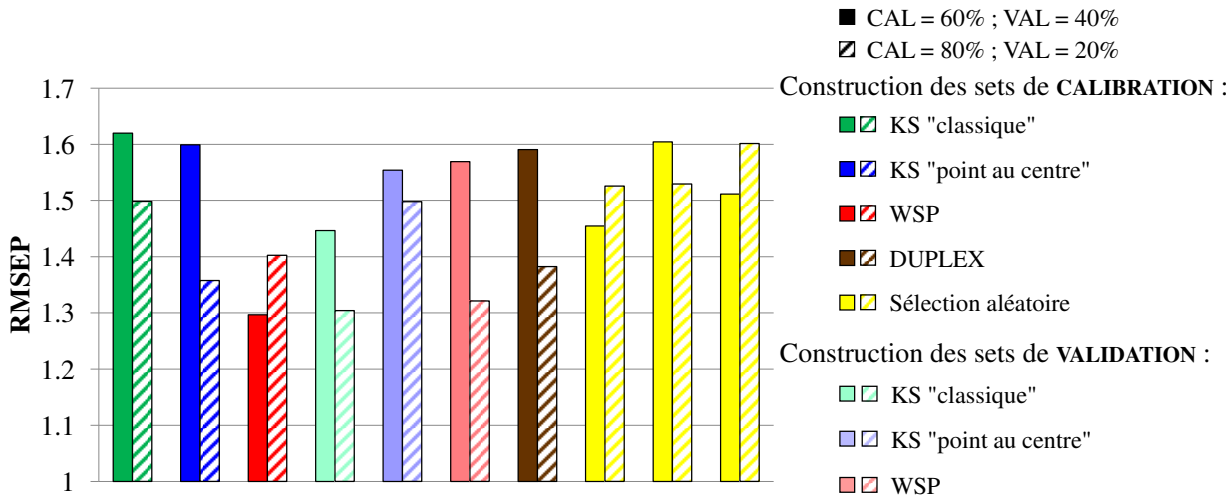


FIGURE 2.70 – RMSEP obtenus à partir des sous-ensembles de validation pour l'analyse de la réponse "Y1". Les critères représentés sont calculés à partir des sous-ensembles de validation contenant 40% et 20% des données.

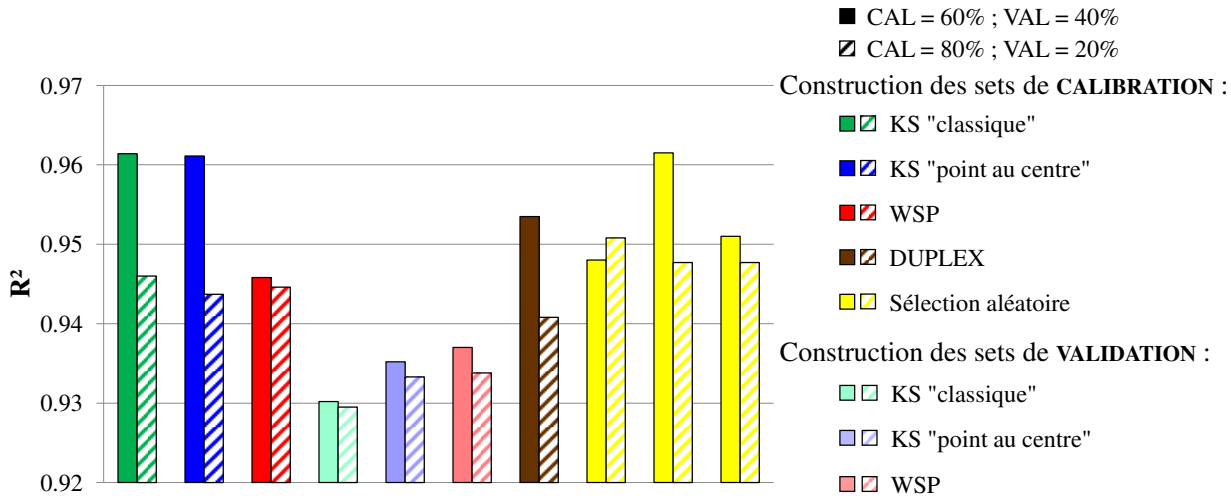


FIGURE 2.71 – R^2 calculés à partir des sous-ensembles de calibration pour l'analyse de la réponse "Y1" comptant 60% ou 80% des données initiales.

Si nous comparons les valeurs des critères *a posteriori* pour les deux répartitions des données, nous observons plus de variations entre les différentes stratégies envisagées lorsque les sets de calibration et de validation comptent respectivement 60% et 40% des données initiales plutôt que la répartition 80%/20%. D'une manière générale ce changement de proportion entraine une diminution du RMSEC et une augmentation du RMSEP. Néanmoins, même si les valeurs de ces critères sont moins "bonnes" lorsque le set de calibration est construit par l'algorithme WSP, nous observons que seule cette stratégie permet d'obtenir des valeurs de RMSEC et RMSEP équivalentes, ce qui caractérise un modèle performant. Par ailleurs, les trois sélections aléatoires du set de calibration présentent plus de fluctuations dans les critères que celles obtenues pour une répartition 80%/20%.

A partir de ces résultats, nous avons pu mettre en évidence qu'un changement du pourcentage de données affectées aux sets de calibration et de validation s'accompagne d'une modification des critères

a posteriori. Ceci laisse supposer que cette étude pourrait être effectuée avec d'autres pourcentages pour la répartition des données afin de trouver le meilleur compromis entre la répartition des données et la qualité des modèles de régression PLS obtenus.

— D'autre part, nous avons exposé le principe du "repliage" qui consiste à projeter les points d'un plan d'expériences initial de dimension D dans le sous-espace des variables détectées comme les plus influentes. Nous avons montré que le plan d'expériences, initialement uniforme, ne présente pas systématiquement des propriétés de recouvrement uniforme dans le sous-espace de projection. Il serait donc intéressant d'élaborer des méthodes de construction de Space Filling Designs, qui garantissent *a priori* l'uniformité des plans projetés dans tous les sous-espaces, quelles que soient les variables retenues lors de l'analyse de sensibilité préalable. Nous pourrions alors considérer ces plans d'expériences comme des "plans universels", constitués d'un ensemble de points de qualité optimale pour une étape de screening et qui demeureraient de qualité optimale pour une étape de surface de réponse ultérieure dans le sous-espace des variables influentes et ce, quelles que soient les dimensions de départ et d'arrivée.

Ces quelques idées semblent intéressantes pour parfaire le catalogue des outils adaptés à la grande dimension mais méritent d'être approfondies dans des travaux futurs.

Bibliographie

[1] H. Niederreiter, "Point sets and sequences with small discrepancy," *Monatshefte für Mathematik*, vol. 104, no. 4, pp. 273–337, 1987.

[2] H. Niederreiter, "Low-discrepancy and low-dispersion sequences," *Journal of Number Theory*, vol. 30, no. 1, pp. 51–70, 1988.

[3] P. Audze and V. Eglais, "New approach for planning out of experiments," *Problems of dynamics and strengths*, vol. 35, pp. 104–107, 1977.

[4] S. J. Bates, J. Sienz, and D. S. Langley, "Formulation of the audze–eglais uniform latin hypercube design of experiments," *Advances in Engineering Software*, vol. 34, no. 8, pp. 493–506, 2003.

[5] M. Johnson, L. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, vol. 26, no. 2, pp. 131–148, 1990.

[6] V. Chen, K. Tsui, R. Barton, and M. Meckesheimer, "A review on design, modeling and applications of computer experiments," *IIE Transactions*, vol. 38, no. 4, pp. 273–291, 2006.

[7] M. Trosset, "Approximate maximin distance designs," in *Proceedings of the Section on Physical and Engineering Sciences*, pp. 223–227, 1999.

[8] M. Gunzburger and J. Burkhardt, "Uniformity measures for point samples in hypercubes," 2004.

[9] C. A. Bennett and N. L. Franklin, *Statistical analysis in chemistry and the chemical industry*. Wiley publications in statistics, New York : Wiley, 1954.

[10] J. P. Benzécri, *L'analyse des données : L'analyse des correspondances*. Dunod, 1973.

[11] T. Cox and M. A. A. Cox, *Multidimensional Scaling, Second Edition*. CRC Press, 2000.

[12] P. Demartines, *Analyse de données par réseaux de neurones auto-organisés*. Thesis, Institut National Polytechnique de Grenoble, France, 1994.

[13] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling : Theory and Applications*. Springer Science & Business Media, 2005.

[14] J. Tenenbaum, V. De Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[15] K. Weinberger, B. Packer, and L. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[16] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.

[17] J. Li, "Visualization of high-dimensional data with relational perspective map," *Information Visualization*, vol. 3, no. 1, pp. 49–59, 2004.

[18] K. Weinberger and L. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 77–90, 2006.

[19] H. Chen, G. Jiang, and K. Yoshihira, "Robust nonlinear dimensionality reduction for manifold learning," in *18th International Conference on Pattern Recognition, 2006. ICPR 2006*, vol. 2, pp. 447–450, 2006.

[20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[21] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms, Second Edition*. Cambridge, Mass : The MIT Press, 2nd ed., 2001.

[22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[23] L. Saul and S. Roweis, "An introduction to locally linear embedding," *unpublished. Available at : http ://www. cs. toronto. edu/˜ roweis/lle/publications. html*, 2000.

[24] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[25] T. Kohonen, *Self-Organizing Maps*. Berlin ; New York : Springer, 3rd ed., 2000.

[26] C. Bishop, M. Svensén, and C. Williams, "GTM : The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215–234, 1998.

[27] P. Demartines and J. Herault, "Curvilinear component analysis : a self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, 1997.

[28] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, no. 1, pp. 137–148, 1969.

[29] M. Daszykowski, B. Walczak, and D. L. Massart, "Representative subset selection," *Analytica Chimica Acta*, vol. 468, no. 1, pp. 91–103, 2002.

[30] R. Snee, "Validation of regression models : Methods and examples," *Technometrics*, vol. 19, no. 4, pp. 415–428, 1977.

[31] E. Marengo and R. Todeschini, "A new algorithm for optimal, distance-based experimental design," *Chemometrics and Intelligent Laboratory Systems*, vol. 16, no. 1, pp. 37–44, 1992.

[32] R. Clark, "OptiSim : An extended dissimilarity selection method for finding diverse representative subsets," *Journal of Chemical Information and Computer Sciences*, vol. 37, no. 6, pp. 1181–1188, 1997.

[33] 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, Marseille, France, 1989.

[34] M. Sergent, R. Phan-Tan-Luu, and J. Elguero, "Statistical analysis of solvent scales, part 1," *Anales de Química Int. Ed. 93*, no. 2, pp. 71–75, 1997.

[35] M. Sergent, R. Phan-Tan-Luu, R. Faure, and J. Elguero, "Statistical analysis of solvents scales, part 2," *Anales de Química Int. Ed. 93*, no. 5, pp. 295–300, 1997.

[36] J. Santiago, M. Claeys-Bruno, and M. Sergent, "Construction of space-filling designs using WSP algorithm for high dimensional spaces," *Chemometrics and Intelligent Laboratory Systems*, vol. 113, pp. 26–31, 2012.

[37] A. Beal, J. Santiago, M. Claeys-Bruno, and M. Sergent, "Repairing uniform experimental designs : Detection and/or elimination of clusters, filling gaps," *Chemometrics and Intelligent Laboratory Systems*, vol. 134, pp. 140–147, 2014.

[38] G. Puchwein, "Selection of calibration samples for near-infrared spectrometry by factor analysis of spectra," *Analytical Chemistry*, vol. 60, no. 6, pp. 569–573, 1988.

[39] P. Mahalanobis, "On the generalized distance in statistics," vol. 2, pp. 49–55, 1936.

[40] R. De Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.

[41] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226–231, AAAI Press, 1996.

[42] M. Daszykowski, B. Walczak, and D. L. Massart, "Looking for natural patterns in data : Part 1. density-based approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 56, no. 2, pp. 83–92, 2001.

[43] J. MacQueen, "Some methods for classification and analysis of multivariate observations," The Regents of the University of California, 1967.

[44] D. Massart and L. Kaufman, *The interpretation of analytical chemical data by the use of cluster analysis*. Wiley, 1983.

[45] W. Vogt, D. Nagel, and H. Sator, *Cluster analysis in clinical chemistry : a model*. Wiley, 1987.

[46] R. Bellman, *Dynamic programming*. Princeton University Press, 1957.

[47] M. Köppen, "The curse of dimensionality," in *5th Online World Conference on Soft Computing in Industrial Applications*, pp. 4–8, 2000.

[48] A. Beal, M. Claeys-Bruno, and M. Sergent, "Constructing space-filling designs using an adaptive WSP algorithm for spaces with constraints," *Chemometrics and Intelligent Laboratory Systems*, vol. 133, pp. 84–91, 2014.

[49] I. T. Jolliffe, "Discarding variables in a principal component analysis. i : Artificial data," *Applied Statistics*, vol. 21, no. 2, pp. 160–173, 1972.

[50] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[51] H. Kaiser, "The application of electronic computers to factor analysis," *Educational and Psychological Measurement*, vol. 20, pp. 141–151, 1960.

[52] R. Todeschini, V. Consonni, and A. Maiocchi, "The k correlation index : theory development and its application in chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 46, no. 1, pp. 13–29, 1999.

[53] D. Livingstone and E. Rahr, "Corchop – an interactive routine for the dimension reduction of large QSAR data sets," *Quantitative Structure-Activity Relationships*, vol. 8, no. 2, pp. 103–108, 1989.

[54] R. Todeschini, D. Ballabio, V. Consonni, A. Manganaro, and A. Mauri, "Canonical measure of correlation (CMC) and canonical measure of distance (CMD) between sets of data. part 1 : Theory and simple chemometric applications," *Analytica Chimica Acta*, vol. 648, no. 1, pp. 45–51, 2009.

[55] V. Consonni, D. Ballabio, A. Manganaro, A. Mauri, and R. Todeschini, "Canonical measure of correlation (CMC) and canonical measure of distance (CMD) between sets of data. part 2 : Variable reduction," *Analytica Chimica Acta*, vol. 648, no. 1, pp. 52–59, 2009.

[56] B. F. Green, "The orthogonal approximation of an oblique structure in factor analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.

[57] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[58] D. Kendall, "A survey of the statistical theory of shape," *Statistical Science*, vol. 4, no. 2, pp. 87–99, 1989.

[59] J. Andrade, M. Gomez-Carracedo, W. Krzanowski, and M. Kubista, "Procrustes rotation in analytical chemistry, a tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, no. 2, pp. 123–132, 2004.

[60] J. N. R. Jeffers, "Two case studies in the application of principal component analysis," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 16, no. 3, pp. 225–236, 1967.

[61] J. Tukey, *Exploratory data analysis*, vol. 231. 1977.

[62] R. McGill, J. Tukey, and W. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[63] C. Selassie, R. Verma, and D. Abraham, "History of quantitative structure–activity relationships," in *Burger's Medicinal Chemistry and Drug Discovery*, pp. 1–96, John Wiley & Sons, Inc., 2003.

[64] S. Benson and J. Buss, "Additivity rules for the estimation of molecular properties. thermodynamic properties," *The Journal of Chemical Physics*, vol. 29, pp. 546–572, 1958.

[65] M. Karelson, V. Lobanov, and A. R. Katritzky, "Quantum-chemical descriptors in QSAR/QSPR studies," *Chemical reviews*, vol. 96, no. 3, pp. 1027–1044, 1996.

[66] R. Pearson, "Hard and soft acids and bases," *Journal of the American Chemical Society*, vol. 85, no. 22, pp. 3533–3539, 1963.

[67] C. Caro, J. Zagal, F. Bedioui, C. Adamo, and G. Cárdenas-Jirón, "Solvent effect on density functional reactivity indexes applied to substituted nickel phthalocyanines," *The Journal of Physical Chemistry A*, vol. 108, no. 28, pp. 6045–6051, 2004.

[68] G. Cárdenas-Jirón, S. Gutiérrez-Oliva, J. Melin, and A. Toro-Labbé, "Relations between potential energy, electronic chemical potential, and hardness profiles," *The Journal of Physical Chemistry A*, vol. 101, no. 25, pp. 4621–4627, 1997.

[69] R. Mulliken, "A new electroaffinity scale ; together with data on valence states and on valence ionization potentials and electron affinities," *The Journal of Chemical Physics*, vol. 2, no. 11, pp. 782–793, 1934.

[70] G. Montaudo, M. Montaudo, and F. Samperi, "Matrix-assisted laser desorption/ionization mass spectrometry of polymers (MALDI-MS)," in *Mass spectrometry of polymers*, pp. 419–522, CRC Press, 2001.

[71] H. Pasch and W. Schrepp, *MALDI-TOF Mass Spectrometry of Synthetic Polymers*. Berlin ; New York : Springer, 2003.

[72] J. Scrivens and A. Jackson, "The effect of the variation of cation in the matrix-assisted laser desorption/ionisation-collision induced dissociation (MALDI-CID) spectra of oligomeric systems," *International Journal of Mass Spectrometry and Ion Processes*, vol. 165, pp. 363–375, 1997.

[73] "Dragon (software for molecular descriptor calculation)," 2013.

[74] R. Todeschini and V. Consonni, *Molecular Descriptors for Chemoinformatics*, vol. 41. WILEY-VCH ed., 2009.

[75] I. V. Tetko, J. Gasteiger, R. Todeschini, A. Mauri, D. Livingstone, P. Ertl, V. A. Palyulin, E. V. Radchenko, N. S. Zefirov, A. S. Makarenko, V. Y. Tanchuk, and V. V. Prokopenko, "Virtual computational chemistry laboratory–design and description," *Journal of Computer-Aided Molecular Design*, vol. 19, no. 6, pp. 453–463, 2005.

[76] H. Wold, "Partial least squares," in *Encyclopedia of Statistical Sciences*, vol. 9, John Wiley & Sons, Inc., 2004.

[77] M. Tenenhaus, *La régression PLS : théorie et pratique*. Editions TECHNIP, 1998.

[78] G. Mazerolles, M. Hanafi, E. Dufour, D. Bertrand, and E. M. Qannari, "Common components and specific weights analysis : A chemometric method for dealing with complexity of food products," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 41–49, 2006.

[79] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts : The MathWorks Inc., 2010.

[80] K. Fang, D. K. Lin, P. Winker, and Y. Zhang, "Uniform design : Theory and application," *Technometrics*, vol. 42, no. 3, pp. 237–248, 2000.

[81] K. Fang, R. Li, and A. Sudjianto, *Design and modeling for computer experiments*. Chapman & Hall/CRC, 2006.

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

[83] H. Faure, "Discrépances de suites associées à un système de numération (en dimension un)," *Bull. Soc. math. France*, no. 109, pp. 142–182, 1981.

[84] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, no. 1, pp. 84–90, 1960.

[85] J. M. Hammersley, "Monte carlo methods for solving multivariable problems," *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960.

[86] I. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, 1967.

[87] I. Sobol, "Uniformly distributed sequences with an additional uniform property," *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 5, pp. 236–242, 1976.

[88] A. Saltelli, S. Tarantola, and F. Campolongo, "Sensitivity analysis as an ingredient of modeling," *Statistical Science*, vol. 15, no. 4, pp. 377–395, 2000.

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

[90] J. Van der Corput, "Verteilungsfunktionen," *Proceedings of the Koninklike Nederlands Akademie van Wetenschappen*, no. 38, pp. 813–821, 1935.

[91] P. Bratley and B. Fox, "Algorithm 659 : Implementing sobol's quasirandom sequence generator," *ACM Trans. Math. Softw.*, vol. 14, no. 1, pp. 88–100, 1988.

[92] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[93] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[94] B. Tang, "Orthogonal array-based latin hypercubes," *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1392–1397, 1993.

[95] J. Park, "Optimal latin-hypercube designs for computer experiments," *Journal of Statistical Planning and Inference*, vol. 39, no. 1, pp. 95–111, 1994.

[96] F. Xiong, Y. Xiong, W. Chen, and S. Yang, "Optimizing latin hypercube design for sequential sampling of computer experiments," *Engineering Optimization*, vol. 41, no. 8, pp. 793–810, 2009.

[97] A. Owen, "Orthogonal arrays for computer experiments, integration and visualisation," *Statistica Sinica 2*, pp. 439–452, 1992.

[98] K. Ye, "Orthogonal column latin hypercubes and their application in computer experiments," *Journal of the American Statistical Association*, vol. 93, no. 444, pp. 1430–1439, 1998.

[99] 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.

[100] J. Franco, X. Bay, D. Dupuy, and B. Corre, "Planification d'expériences numériques à partir du processus ponctuel de strauss," 2008.

[101] D. Strauss, "A model for clustering," *Biometrika*, vol. 62, no. 2, pp. 467–475, 1975.

[102] A. Saltelli, S. Tarantola, and K. Chan, "A quantitative model-independent method for global sensitivity analysis of model output," *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.

[103] B. Iooss, "Revue sur l'analyse de sensibilité globale de modèles numériques," *Journal de la Société Française de Statistique*, vol. 152, no. 1, pp. 3–25, 2011.

[104] A. Saltelli and I. Sobol, "About the use of rank transformation in sensitivity analysis of model output," *Reliability Engineering & System Safety*, vol. 50, no. 3, pp. 225–239, 1995.

[105] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice : A Guide to Assessing Scientific Models*. Hoboken, NJ : Wiley-Blackwell, 2004.

[106] R. Todeschini, D. Ballabio, V. Consonni, F. Sahigara, and P. Filzmoser, "Locally centred mahalanobis distance : A new distance measure with salient features towards outlier detection," *Analytica Chimica Acta*, vol. 787, pp. 1–9, 2013.