

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
UNIVERSITE BLIDA 1



FACULTE DES SCIENCES
DEPARTEMENT DE MATHÉMATIQUES

MEMOIRE DE MASTER

En Mathématiques

Spécialité : Recherche Opérationnelle

Thème

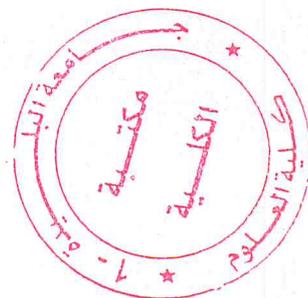
Plans D-Optimaux

Algorithmes d'échange et de déplacement

Présenté par :
AZAZI Sabrina

Devant le jury composé de :

Président : Mr O. TAMI
Examineurs : Mr F.HANNANE
Promoteur : Mr H. EL MOSSAOUI

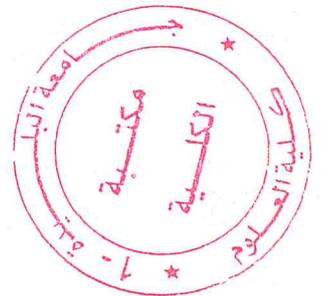


ملخص

ان طريقة المنهج التجريبي هي مجموعة من الطرق و وسائل للتحليل يستعملها الباحثون من اجل تخطيط تجاربهم. الهدف من الطريقة السماح للباحثين تجويد فعالية تجاربهم مهما كان ميدان تخصصهم كما يساعد على تحليل المشكل لطرح إستراتيجية بحتة ذات جودة عالية بدلالة الأهداف المسطرة و الوسائل المتوفرة لهذه المنهجية.

لهذا العمل هدفان :

- دراسة خوارزميات التبادل و خوارزميات عن طريق الانتقال من ايجاد احسن مصفوفة و مقارنة المصفوفات لتوجيه اختيار المحرب.
- دراسة جديدة في خوارزميات Donev التي تتمثل في بعض التغييرات مع تحديد سلبياتها و ايجابياتها و مقارنتها مع خوارزمية Donev الاصلية .

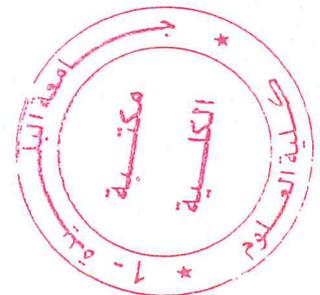


Résumé

La méthodologies de la recherche expérimentale (MRE) est un ensemble de méthode et de mode de raisonnement destiné à tout expérimentateur désirant faire de la planification expérimentale. Elle a pour objet de lui permettre d'optimiser l'efficacité de sa recherche expérimentale quel que soit sa branche d'activité. Pour ce faire, elle va l'aider à exprimer aux mieux son problème et lui proposer des stratégies expérimentales optimales en fonction des objectifs qu'il s'est fixé et des moyennes dont il dispose.

Ce travail comporte deux objectifs :

- Les algorithmes qui consistent à trouver la meilleure matrice selon le critère D-optimal et il existe deux catégories, les algorithmes d'échanges et ceux par les déplacements, puis on fait une étude comparative sur les deux catégories afin d'aider l'expérimentateur à prendre des décisions.
- Une nouvelle modification sur l'algorithme de Donev tout en précisant ses avantages et ses inconvénients par rapport à l'algorithme de Donev.



Abstract

The methodology of experimental research (MRE) is a set of methods and fashion reasoning for any experimenter wishing to make the planning experimental planification. Its purpose is to allow him to optimize effectiveness of its experimental research regardless of its branch activity. To do this, it will help to better express his problem and propose optimal experimental strategies based on objectives that It is fixed and mean it has.

This work has two objectives:

- The algorithms are to find the best matrix according to the D-criterion optimal and there are two categories of algorithms by those exchanges and investments, then a comparative study of two categories to help the experimenter to make decisions.
- A new modification of Donev algorithm while stating its advantages and disadvantages compared to Donev algorithm.

Remerciements

Je remercie Dieu, le tout puissant, qui nous a donné le courage pour accomplir ce travail.

Nos vifs remerciements à tous ceux qui nous ont aidées de près ou de loin pour la réalisation de ce modeste travail. surtout mon ami I.DIA.

Nous présentons nos sincères remerciements en particulier à nos promoteurs : monsieur Hannane Farouk et monsieur Elmousaoui Hichem pour nous avoir honorées en acceptant de diriger ce travail. Ses conseils et orientations nous ont beaucoup aidés. Nous tenons à noter tout le plaisir que nous avons eu à travailler avec eux. Sans oublier de remercier monsieur le chef de département : Tami Omar ainsi que nos respectables professeurs et le personnel de la direction.

Merci....

Dédicace

Je dédie ce mémoire à :

*Mon cher **père** et ma chère **mère** qui ont éclairé mon chemin et qui m'ont encouragé et soutenu tout au long de mes études.*

*Mes très chers **frères** et mes **sœurs**.*

*Mon très cher fiancé **Mohamed** et à tout sa famille **CHAKER**.*

*Toute ma famille paternelle **AZAZI** et maternelle **MEBARKI**.*

*Mes très chers **amis** en particulier **I.DIA** que j'apprécie tellement qui ont été toujours à mon côté pour m'encourager et me donner la force pour continuer.*

*Et tout les enseignants du département de mathématique et a leur tête monsieur **Omar Tami** et Mes chers professeurs **Mr Hannane** et monsieur*

***El moussaoui**.*

Sabrina...

Table de matières

Principaux notions et symboles.....	3
Liste des algorithmes	4
INTRODUCTION GENERALE.....	5
I- NOTIONS DE PLANS D'EXPERIENCES ET CRITERES D'OPTIMALITE	
1- Introduction aux plans d'expériences.....	6
2- Critères d'optimalite.....	10
3- Plans D-optimaux	13
4- Exemple d'un plan D-optimal	14
II- CONSTRUCTION ALGORITHMIQUE DES PLANS D- OPTIMAUX PAR LES ALGORITHMES D'ECHANGE	
1. Introduction aux plan D-optimaux.....	16
2. Quelques algorithmes d'échanges.....	17
2.1- Algorithme de Fedorov 1972.....	19
2.2- Algorithme de Fedorov modifié.....	21
2.3- Algorithme de k-échange.....	22
2.4- Algorithme de kl-échange.....	24
2.5- Algorithme de kl-échange modifié.....	26
2.6- Algorithme de Fdop.....	29
3. Application des algorithmes d'échanges sur la fabrication pharmaceutique d'un médicament bien protégé.....	31
3.1- Préparation de l'expérimentation.....	31
3.2- Description des contraintes.....	31
3.3- Choix des facteurs.....	32
3.4- Domaine d'étude.....	32
3.5- Choix des réponses.....	32
3.6- Choix du plan expérimental.....	32
4- Comparaison entre les algorithmes d'échanges.....	34
III- CONSTRUCTION ALGORITHMIQUE DES PLANS D- OPTIMAUX PAR DEPLACEMENT	
1- Les algorithmes de déplacement.....	36
1.1- Algorithme de DONEV.....	36

1.2- Algorithme de Fdop modifié.....	38
2- Application des algorithmes de déplacement sur l'étude explosive.....	39
2.1- Description de la composition explosive.....	39
2.2- Choix des facteurs.....	40
2.3- Domaine d'étude.....	40
2.4- Choix des réponses.....	40
2.5- Choix du plan expérimental.....	41
2.6- Modèle postulé.....	41
2.7- Calcul du plan D-optimal.....	43
3- Comparaison des algorithmes.....	43
IV- LES NOUVELLES PROPOSITIONS	
1- Principe de l'algorithme.....	45
2- Algorithme proposé.....	45
3- Etude comparative	47
CONCLUSION GENERALE.....	54
Annexe.....	55

Symboles

k	Nombre des facteurs.
N	Nombre d'essais d'un plan d'expériences.
n	Nombre de points souhaités.
p	Nombre de coefficients du modèle mathématique.
χ	Domaine expérimental d'intérêt.
E	Ensemble des matrices d'expériences.
ξ	Matrice d'expériences.
ξ_N	Matrice d'expériences à N expériences.
X	Matrice des effets.
y	Vecteur des réponses mesurées.
a_0	Constant du modèle.
a_i	Coefficients des termes de premier degré.
a_{ii}	Coefficients des termes carrés.
a_{ij}	Coefficients des termes rectangles.
a	Vecteur des coefficients du modèle.
e	Vecteur des résidus.
$X'X$	Matrice d'information.
$(X'X)^{-1}$	Matrice de dispersion.
σ	Ecart-type.
σ^2	Variance.
$\hat{\sigma}^2$	Variance de regression.
$\hat{\sigma}_u^2$	Variance de regression au point u .
α	Le risque.
λ_i	Les valeurs propres de la matrice de dispersion.
ξ^*	La matrice optimale des effets.
$x_{(i)}$	Point mauvais.
$x_{(j)}$	Point bon.
$\xi_n^{(0)}$	Matrice initiale des effets à n expériences.
$\xi_n^{(t)}$	Matrice d'effet à l'instant t .
$(X_n^t, X_n)^{(t)}$	Matrice d'information à l'instant t .
$d(\xi_n^{(t)}, x_{(i)})$	Fonction d'erreur de prédiction au point $x_{(i)}$.
$d(\xi_n^{(t)}, x_{(i)}, x_{(j)})$	Fonction d'erreur de prédiction pour le couple $(x_{(i)}, x_{(j)})$.
$\Delta(\xi_n^{(t)}, x_{(i)}, x_{(j)})$	Fonction Δ pour le couple $(x_{(i)}, x_{(j)})$.
ε_{fed}	Maximum de Δ
$maxi$	Nombre d'excursion.
α	Amplitude des déplacements.

Liste des algorithmes

Algorithme de Fedorov	18
Algorithme de Fedorov modifié	19
Algorithme de k-échange	21
Algorithme de kl-échange	22
Algorithme de kl-échange modifié	24
Algorithme de Fdop	25
Algorithme de Donev	31
Algorithme de Fdop modifié	32
Algorithme de Donev modifié	37

INTRODUCTION

Les plans d'expériences font partie des statistiques. Ils constituent actuellement un outil indispensable pour les chercheurs industriels lorsqu'ils abordent l'étude d'un système quelconque (en mécanique, en chimie, en pharmacie, etc...).

Il existe plusieurs types de plans d'expériences, parmi eux, les plans D-optimaux. Ces plans sont souvent utilisés lorsque le domaine expérimental n'est pas entièrement accessible. Il existe des contraintes qui interdisent l'accès à certaines régions du domaine d'étude. Par exemple, il est impossible de réaliser certaines combinaisons de niveaux par manque de matériel ou il existe des dangers liés au niveau des facteurs (explosion, toxicité) ou pour toutes autres raisons. Ces plans permettent d'explorer la plus grande partie possible du domaine expérimental malgré les restrictions dues aux contraintes. Ils sont utilisés aussi lorsque l'on veut diminuer le nombre des essais d'un plan classique. Ils nécessitent l'utilisation d'un logiciel spécifique basé sur les algorithmes d'échange et de déplacement, dont les principaux sont Mitchell et Fedorov.

Dans notre travail, nous utiliserons comme outil informatique le logiciel **MATLAB**.

Dans le premier chapitre, nous rappellerons les principes de la méthode des plans d'expériences, son intérêt ainsi que l'importance du choix des points expérimentaux. Les notions de critères d'optimalités seront abordées. On donnera enfin un exemple de plan D-optimal.

Le deuxième chapitre sera consacré à la construction de plans D-optimaux à l'aide des algorithmes d'échanges. On donnera aussi un exemple de fabrication pharmaceutique dans lequel les algorithmes d'échange seront appliqués et les résultats comparés.

Les algorithmes de déplacement seront étudiés au chapitre trois, un exemple d'étude explosive étant donné à la fin.

L'essentiel de notre travail sera traité au chapitre quatre où nous donnerons un nouvel algorithme (Donev modifié).

Chapitre 1

Notions de plans d'expériences et critères d'optimalité

1. INTRODUCTION AUX PLANS D'EXPERIENCES

Les plans d'expériences sont utilisés dans les études industrielles en recherche-développement. Ils interviennent dans de nombreux domaines industriels. On peut notamment citer :

- industrie chimique, pétrochimique et pharmaceutique.
- industrie mécanique et automobile.
- industrie métallurgique.

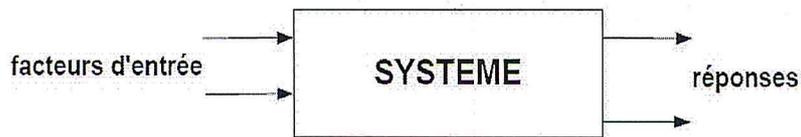
Leur utilisation vise aux buts suivants :

- détermination des facteurs clés dans la conception d'un nouveau produit ou d'un nouveau procédé.
- optimisation des réglages d'un procédé de fabrication ou d'un appareil de mesure.
- prédiction par modélisation du comportement d'un procédé.

Les plans d'expériences s'inscrivent dans une démarche générale d'amélioration de la qualité.

La démarche originale des plans d'expériences réside dans la possibilité d'interprétation de résultats expérimentaux avec un effort minimal sur le plan expérimental : la minimisation du nombre nécessaire d'expériences permet un gain de temps et de coût financier [1].

Le scientifique est souvent amené à comprendre comment réagit un système en fonction des facteurs susceptibles de le modifier. Pour visualiser cette évolution, il mesure une réponse et va ensuite essayer d'établir des relations de cause à effet entre les réponses et les facteurs.



Les facteurs étudiés dans un plan d'expériences sont bien entendus les facteurs d'entrée. Un facteur est une grandeur, le plus souvent mesurable mais il peut s'agir d'une grandeur qualitative comme les différents lots d'une matière première.

La réponse est la grandeur mesurée à chaque essai; le plan vise à déterminer quels facteurs l'influencent ou quelle est son évolution en fonction de ceux-ci. Cette grandeur est le plus souvent mesurable mais elle peut également être qualitative. Dans ce cas, par exemple une appréciation visuelle sur l'état d'une surface ou une appréciation bon, moyen ou mauvais sur un produit alimentaire.

Le succès des plans d'expériences réside dans la diminution du nombre d'essais pour obtenir un maximum d'informations. Néanmoins il ne suffit pas d'effectuer un certain nombre d'essais : le choix des essais est fondamental pour l'obtention d'une précision optimale dans la détermination des coefficients.

Dans l'exemple suivant : on va comparer la précision des estimations de coefficients pour 3 différentes matrices d'expériences ; autrement dit on calculera la variance de chaque coefficient.

La pression, la température et la masse catalyseur sont les facteurs examinés pour la réponse qui est le rendement d'une réaction : 4 essais sont réalisés avec 2 niveaux par facteur. On définit alors les matrices d'expériences des plans 1, 2 et 3 avec la colonne de la réponse (le rendement). On envisage le modèle suivant pour chacun des plans.

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 ,$$

Où x_1 , x_2 et x_3 représentent respectivement les variables représentatives des facteurs pression, température et masse du catalyseur.

N° d'essai	Pression	température	Masse catalyseur	réponse
1	1	1	1	Y ₁
2	-1	1	1	Y ₂
3	1	-1	1	Y ₃
4	1	1	-1	Y ₄
Niveau -1	2bars	50°C	1kg	
Niveau +1	4bars	70°C	2kg	
Niveau 0	3bars	60°C	1,5kg	

Tableau 1 : plan 1

N° d'essai	Pression	Température	Masse catalyseur	réponse
1	-1	-1	1	Y ₅
2	1	-1	-1	Y ₆
3	-1	1	-1	Y ₇
4	1	1	1	Y ₈
Niveau -1	2bars	50°C	1kg	
Niveau +1	4bars	70°C	2kg	
Niveau 0	3bars	60°C	1,5kg	

Tableau 2 : plan 2

N° d'essai	Pression	Température	Masse catalyseur	réponse
1	0	0	0	Y ₉
2	1	1	0	Y ₁₀
3	1	0	1	Y ₁₁
4	0	1	1	Y ₁₂
Niveau -1	2bars	50°C	1kg	
Niveau +1	4bars	70°C	2kg	
Niveau 0	3bars	60°C	1,5kg	

Tableau 3 : plan 3

Pour les coefficients, on aboutit aux solutions suivantes

$$\text{plan 1 : } \begin{cases} a_0 = \frac{-Y_1 + Y_2 + Y_3 + Y_4}{2} \\ a_1 = \frac{Y_1 - Y_2}{2} \\ a_2 = \frac{Y_1 - Y_3}{2} \\ a_3 = \frac{Y_1 - Y_4}{2} \end{cases} \quad \text{plan 2 : } \begin{cases} a_0 = \frac{+Y_5 + Y_6 + Y_7 + Y_8}{4} \\ a_1 = \frac{-Y_5 + Y_6 - Y_7 + Y_8}{4} \\ a_2 = \frac{-Y_5 - Y_6 + Y_7 + Y_8}{4} \\ a_3 = \frac{+Y_5 - Y_6 - Y_7 + Y_8}{4} \end{cases}$$

$$\text{plan 3 : } \begin{cases} a_0 = Y_9 \\ a_1 = \frac{-Y_9 + Y_{10} + Y_{11} - Y_{12}}{2} \\ a_2 = \frac{-Y_9 + Y_{10} - Y_{11} + Y_{12}}{2} \\ a_3 = \frac{-Y_9 - Y_{10} + Y_{11} + Y_{12}}{2} \end{cases}$$

La variance $var(Y)$ de chacune des réponses Y_i est considérée comme identique en chaque point du domaine. Cette hypothèse est couramment admise dans les études de plans d'expériences. Selon les plans on obtient des variances $var(a_i)$ différentes pour les coefficients a_i du modèle.

Plan 1 : $var(a_0) = var(Y)$ et $var(a_1) = var(a_2) = var(a_3) = var(Y) / 2$

Plan 2 : $var(a_0) = var(a_1) = var(a_2) = var(a_3) = var(Y) / 4$

Plan 3 : $var(a_0) = var(a_1) = var(a_2) = var(a_3) = var(Y)$

Remarque : On rappelle que si X et Y sont 2 variables aléatoires indépendantes et α un réel $var(\alpha X) = \alpha^2 var(X)$ et $var(X \pm Y) = var(X) + var(Y)$

On constate donc que le plan 2 présente la détermination la plus précise des coefficients. Intuitivement on peut comprendre que le plan 2 est supérieur car chacun des coefficients est estimé à l'aide de toutes les réponses et les points du domaine expérimental sont bien mieux répartis. Le plan 2 est donc bien préférable pour une estimation précise du modèle.

Ceci montre donc que la matrice d'expérience n'est pas quelconque. elle doit être choisie suivant des critères précis [1].

2. CRITERES D'OPTIMALITE

Pour un type de modèle donné, on recherchera donc le placement optimal des points d'expériences pour lequel l'erreur sur les réponses prédites est la plus faible possible. On gardera à l'esprit que dans tous les cas un critère d'optimalité ne s'applique que pour une situation bien définie, c'est-à-dire pour laquelle on a fixé :

- Le nombre d'expériences du plan,
- Le modèle d'interpolation choisi.

En réalité, il existe plusieurs types d'objectifs finaux recherchés qui s'avèrent être très proches dans leurs finalités. En particulier, on peut ainsi être amené à :

- Minimiser la somme des variances des coefficients,
- Minimiser la variance globale de tous les coefficients,
- Minimiser la plus grande variance de coefficient,
- etc.

E étant l'ensemble des matrices d'expériences exactes constituées de N expériences, un critère est une mesure $C(\xi)$ sur une matrice d'expériences ξ appartenant à E , cette dernière prend des valeurs réelles ou booléennes (oui/non), dans le premier cas le critère est dit quantitatif, dans le second qualitatif.

Pour un critère quantitatif, nous recherchons généralement la valeur la plus petite possible; dans certains cas cependant, nous sommes intéressés par la valeur la plus grande.

Un plan optimal est un plan qui fournit les estimations des effets avec la meilleure précision possible. Nous adapterons aussi qu'un problème de plan optimale peut être défini comme le choix d'un plan à n point pour optimiser un critère de plan spécifié.

Les propriétés des plans optimaux font qu'ils sont toujours un bon choix, qu'il s'agisse de criblage ou de modélisation. Ils permettent à l'expérimentateur d'avoir le plan sur mesure qui s'adapte parfaitement aux contraintes de l'étude. Les plans optimaux permettent également de traiter des situations difficiles et sont parfois la seule solution aux problèmes posés. On les emploie lorsqu'il faut :

- Diminuer le nombre d'essais d'un plan classique pour s'adapter au budget et aux délais imposés,
- Adapter le plan aux objectifs de l'étude et aux moyens de l'expérimentateur,
- Traiter en même temps des facteurs de mélanges et des facteurs de procédés,
- Obtenir une surface de réponse en présence de facteurs discrets,
- Tenir compte des contraintes qui pèsent sur le domaine d'étude, toutes les combinaisons de facteurs ne sont pas possibles et le domaine d'étude n'est plus un carré, un cube ou un hyper cube,
- compléter par quelques essais supplémentaires une expérimentation mal conduite afin d'en tirer des conclusions valides (réparation d'expérimentation)

Plusieurs critères de plan populaires (appelées aussi critères d'optimalité) sont discutés dans le paragraphe suivant.

L'approche traditionnelle utilisée pour résoudre un problème de plan optimal consiste à sélectionner les n points candidats à partir d'un plan S de N points. Toutefois, certaines approches d'ensemble non-candidats ont été présentées dans la littérature. Le modèle linéaire pour les m facteurs (ou composants dans le cas des expériences de mélange) dans n itérations dans la matrice de notation est $E(y) = X\beta$, où y est le vecteur d'observations ($n \times 1$), X est une matrice des termes du modèle ($n \times p$) où $p \geq m$ qui contient n vecteurs lignes de termes du modèle x'_i (obtenus à partir du point de plan x_i), et β vecteur des coefficients modèle de régression ($p \times 1$). En admettant que $X'X$ est non singulière, plusieurs critères d'optimalité commune consistent à réduire au minimum les différentes fonctions de $(X'X)^{-1}$.

D-optimalité : Minimiser $|X'X|^{-1}$ (Ou maximiser $|X'X|$) résultats en minimisant la variance des coefficients estimés et réduire au minimum le volume de l'ellipsoïde de confiance (En supposant que les erreurs sont normalement distribuées).

A-optimalité : Minimiser la trace de $|X'X|^{-1}$ résultats en minimisant la variance moyenne des coefficients estimés.

G-optimalité : Minimiser $v_{max} = \max\{v_i\}$ résultats en minimisant au maximum la variance de prédiction normalisée, où $v_i = x'_i (X'X)^{-1} x_i$ est la variance de prédiction normalisée par σ^2 au point expérimental x_i . Idéalement, v_{max} doit être

calculé sur l'ensemble de la région expérimentale. Cependant, de nombreux logiciels de génération de plan qui repose sur un ensemble de points candidats calculent seulement v_{max} sur cet ensemble.

E-optimalité : Une matrice d'expériences est dite E-optimale si la valeur propre maximale de $(X'X)^{-1}$ est la plus faible possible.

R-optimalité : Pour une matrice d'expériences ξ_N , le critère-R est défini par $R(\xi_N) = N$, la minimisation de ce critère n'est pas une fin en soi. Il s'agit plutôt de rechercher le nombre minimal d'expériences permettant d'atteindre les objectifs fixés par l'expérimentateur.

Critère d'orthogonalité : Une matrice d'expériences est orthogonale si elle permet d'obtenir des estimations des coefficients indépendants. Cela se caractérise par des axes de l'ellipsoïde parallèles aux axes des coefficients. Cette propriété est obtenue quand $(X'X)^{-1}$ est diagonale donc quand les covariances des coefficients sont nulles.

Critère de Presque Orthogonalité : Si la matrice $(X'X)^{-1}$ obtenue en retirant sa première ligne et sa première colonne est diagonale, le critère de presque orthogonalité est respecté.

Critère d'isovariance par rotation : le critère d'isovariance par rotation est respecté si l'erreur de prédiction est la même pour des point situés à égale distance du centre du domaine d'étude. On trouve ce critère désigné par rotability dans la littérature anglo-saxonne.

Il existe de nombreux autres critères d'optimalités, souvent dérivés de ceux présentés précédemment. Dans tous les cas, ils se basent sur des considérations de précision des coefficients de modèle ou bien des valeurs estimées par ce modèle.

3. PLANS D-OPTIMAUX

Un plan D-optimal est un plan qui permet non seulement d'étudier les effets des différents facteurs mais aussi il permet de représenter le phénomène par un modèle mathématique quadratique, delà on peut effectuer une optimisation sans contrainte.

Les plans D-optimaux sont une forme de plan fourni par un algorithme informatique. Ces types de plans assistés par ordinateur sont particulièrement utiles lorsque les plans classiques ne sont pas applicables.

Contrairement au plan classique standard tel que les plans factoriels et les plans factoriels fractionnaires, les matrices de plan D-optimaux sont généralement non orthogonales et les estimations de l'effet sont corrélées.

Le critère d'optimalité utilisé dans les modèles D-optimale est une maximisation du déterminant de la matrice d'information $X'X$.

Lorsqu'on calcule l'inverse de la matrice d'information, on est amené à utiliser le déterminant de $X'X$ comme diviseur. Plus la valeur de ce déterminant est grande, plus les éléments de $(X'X)^{-1}$ sont petits. Or c'est ce que nous cherchons puisque ces éléments diagonaux sont proportionnels aux variances des coefficients. Il en résulte que pour un modèle mathématique postulé, le critère de D-optimalité correspond à la disposition des points expérimentaux qui conduit au déterminant de $X'X$ ayant la plus forte valeur possible. On obtient ainsi un plan qui minimise la variance globale de tous les coefficients du modèle.

L'algorithme informatique choisit l'ensemble optimal de plan qui fonctionne à partir d'un ensemble candidat de traitement de plans possibles. Cet ensemble candidat de traitement habituellement se compose de toutes les combinaisons possibles des différents niveaux de facteurs que l'on souhaite utiliser dans l'expérience.

Les raisons d'utiliser les plans D-optimaux au lieu de plans classiques standards se répartissent généralement en deux catégories:

1. l'espace de plan est contraint (l'espace de processus contient les paramètres de facteurs qui ne sont pas réalisables ou impossibles à exécuter),

2. les plans factoriels fractionnaires ou standards exigent trop d'exécutions pour la quantité de ressources ou de temps accordé pour l'expérience.

Il s'applique au cas fréquent des plans sous contraintes. Grâce à ce critère, on conserve un domaine d'étude étendu à tous les essais hors contraintes. Un autre avantage très important de ce critère est la possibilité de choisir avant l'expérimentation le nombre d'essais à réaliser.

Ils sont aussi utilisés lorsque l'on veut diminuer le nombre des essais d'un plan classique. On impose un nombre d'essais et l'algorithme de calcul des plans conserve pour un modèle donné les meilleurs points d'expériences.

4. EXEMPLE D'UN PLAN D-OPTIMAL

Soient x_1, x_2 deux facteurs, pour démontrer l'approche D-optimale de façon géométrique, les deux facteurs sont étudiés sur trois niveaux, $-1, 0$ et 1 . L'ensemble candidat correspondant est mentionné dans le tableau suivant.

Nombre	1	2	3	4	5	6	7	8	9
x_1	-1	-1	-1	0	0	0	1	1	1
x_2	-1	0	1	-1	0	1	-1	0	1

Tableau 4 : Ensemble candidat à deux facteurs avec trois niveaux.

Si nous voulons afficher l'ensemble candidat à la notation des matrices, la matrice ξ_N doit contenir deux colonnes, une pour chaque facteur et $N = 9$ expériences.

Compte tenu d'une conception avec seulement trois courses ($n = 3$) ; nous avons 84 sous-ensembles possibles hors de cet ensemble de candidats. Dans cet exemple, nous évaluons quatre conceptions possibles.

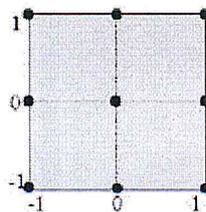


Figure 1 : Répartition des candidats sur la région expérimentale.

L'enquête prend en considération deux facteurs avec trois niveaux et donc neuf points candidats de la matrice et de les comparer en fonction des D-critères. L'équation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$ montre les quatre sous-ensembles sélectionnés dans la notation de la matrice.

$$\xi_{N_{c_1}} = \begin{bmatrix} -1 & -1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \quad \xi_{N_{c_2}} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \xi_{N_{c_3}} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \xi_{N_{c_4}} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}$$

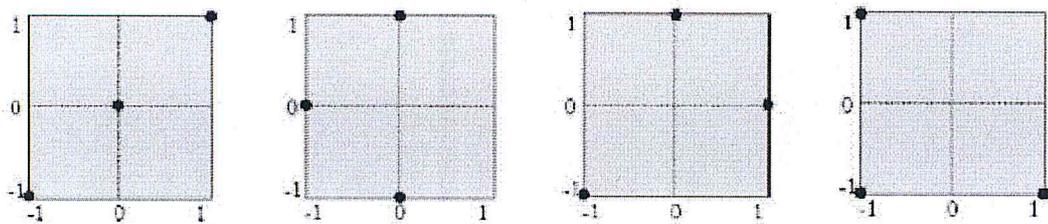


Figure 2 : Répartition de la matrice du modèle.

Pour calculer l'optimalité de ces conceptions, nous devons choisir un modèle linéaire pour que l'exemple soit le plus simple possible :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon,$$

En comparant les conceptions selon le D-critère, n'est possible que si nous calculons le déterminant de la matrice d'information $(X'X)$ pour chacun des quatre motifs.

Expérience	Déterminant
$\xi_{N_{c_1}}$	0
$\xi_{N_{c_2}}$	4
$\xi_{N_{c_3}}$	9
$\xi_{N_{c_4}}$	16

Tableau 5 : Les différents déterminants des expériences.

Chapitre 2

CONSTRUCTION ALGORITHMIQUE DES PLANS D-OPTIMAUX PAR ECHANGE

1. INTRODUCTION AUX PLANS D-OPTIMAUX

La recherche du meilleur plan n'est pas une méthode exacte, mais plutôt une procédure algorithmique utilisant certaines stratégies, pour cela il existe un grand nombre d'algorithmes pour la génération des matrices d'expériences qui utilisent le critère D-optimal.

On a déjà vu que le principe de ce critère consiste à choisir, parmi un grand nombre N de points expérimentaux, les n points expérimentaux qui donnent le déterminant de $X'X$ le plus élevé. On pourrait calculer tous les déterminants possibles et choisir le plus élevé. Le nombre de déterminants à calculer est égal au nombre de combinaisons de N objets pris n à n . Pour 20 essais choisis parmi 100, il faudrait calculer

$$C_{100}^{20} = \frac{100!}{80!20!} \cong 5,36 \times 10^{20}$$

Ce nombre dépasse les capacités des ordinateurs d'aujourd'hui. Si le calcul d'un déterminant demandait un millième de seconde, il faudrait plus de 10 milliards d'années pour mener à bien ce simple problème. On a donc recours à des artifices de calcul utilisant des algorithmes. Ces algorithmes sont basés sur l'échange des points. On commence par définir un nombre élevé de points qui pourraient être inclus dans le plan final et qui tiennent compte des contraintes. Ce sont les points candidats. Il y a N points candidats. Puis, on choisit un modèle mathématique a priori. Enfin, on définit le nombre, n d'essais que l'on désire réaliser. L'algorithme cherche un déterminant élevé en opérant par échange.

L'algorithme choisit, au hasard, n points parmi les N points candidats. C'est le plan initial. Il calcule le déterminant $X'X$ correspondant. L'ordinateur retire un point des n points du plan initial et le remplace par un point des points candidats. Il calcule le

déterminant de la nouvelle matrice $X'X$. Si ce deuxième déterminant est plus élevé que le premier, l'ordinateur conserve le point ; sinon il le rejette et prend un deuxième point parmi les points candidats. L'algorithme poursuit son calcul en échangeant tous les points du plan a n essais avec tous les points candidats. Le calcul se poursuit ainsi jusqu'à ce que la substitution d'un point par un autre ne fasse plus regresser suffisamment la valeur du déterminant.

Nous n'avons pas indiqué tous les détails de ce calcul car ils sont nombreux, complexes et comportent des pièges. Par exemple, il n'est pas équivalent de retirer un point et d'en ajouter un autre ou d'ajouter un point et d'en retirer un autre. On peut tomber sur des maximums locaux et les procédures doivent être envisagées pour s'en dégager . C'est le principe des excursions au cours desquelles l'échange se fait sur 2, 3, 4 et jusqu'à 6 points au lieu d'un. Les principaux algorithmes qui ont fait leur preuve sont l'algorithme d'échange de Fedorov et l'algorithme d'échange de Mitchell. Les logiciels de plans d'expériences possèdent des algorithmes parfois un peu différents, mais qui s'inspirent toujours de la procédure d'échange.

Le résultat du calcul n'est pas unique. Si on recommence plusieurs fois la recherche d'un plan D-optimal, on obtient des propositions de plans qui peuvent être un peu différentes les unes des autres. Il est donc conseillé de faire plusieurs tentatives et de conserver les points expérimentaux qui sont le plus souvent cités. La pratique montre que l'on obtient toujours des plan satisfaisants et que les petites différences constatées ne sont pas gênantes.

2. ALGORITHMES D'ÉCHANGES

Les algorithmes d'échange réduisent le domaine d'intérêt a un sous ensemble fini de points, cette discrétisation du domaine définit le sous ensemble de point que l'expérimentateur est prêt à réaliser. C'est dans ce dernier qu'on trouve la matrice d'expériences à suivre si elle existe.

Ces algorithmes sont une succession de remplacement des mauvais points de la matrice par les bons points pris parmi les points candidats (on dit qu'un point $x^{(i)}$ est mauvais, et le point $x^{(j)}$ est bon si le point $x^{(i)}$ entraîne un déterminant de la matrice de dispersion plus élevé que celui obtenu avec le point $x^{(j)}$). A chaque étape on observe la diminution du déterminant de la matrice de dispersion.

Pour augmenter la probabilité d'atteindre la solution optimale, il est conseillé d'effectuer plusieurs essais en changeant la matrice de départ $\xi_n^{(0)}$. Cette dernière est choisie aléatoirement parmi les points candidats ou fixé par l'expérimentateur à condition que la matrice de dispersion existe ou bien le déterminant de la matrice d'information soit non nul.

La principale différence entre les algorithmes suivants est le processus des échanges de points que l'on rencontre à chaque itération.

Théorème 1: Les matrices d'informations $(X_n^t, X_n)^{(t)}$ et $(X_{n+1}^t, X_{n+1})^{(t+1)}$ sont liées par la relation suivant après l'ajout de l'expérience j qui provoque l'augmentation maximale du déterminant :

$$(X_{n+1}^t, X_{n+1})^{(t+1)} = (X_n^t, X_n)^{(t)} + f(x^{(j)}) \cdot f^t(x^{(j)}) \quad (1)$$

De même les déterminants sont liés par :

$$|X_{n+1}^t, X_{n+1}|^{(t+1)} = |X_n^t, X_n|^{(t)} * |1 + d(\xi_n, x^{(j)})|$$

Avec :

$$d(\xi_n, x^{(j)}) = f^t(x^{(j)}) \cdot ((X_n^t, X_n)^{(t)})^{-1} \cdot f(x^{(j)})$$

Après la suppression de l'expérience i qui provoque la diminution minimale du déterminant, les matrices d'informations $(X_{n+1}^t, X_{n+1})^{(t+1)}$ et $(X_n^t, X_n)^{(t+2)}$ sont liées par la relation:

$$(X_n^t, X_n)^{(t+2)} = (X_{n+1}^t, X_{n+1})^{(t+1)} - f(x^{(i)}) \cdot f^t(x^{(i)}) \quad (2)$$

De même les déterminants sont liés par :

$$|X_n^t, X_n|^{(t+2)} = |X_{n+1}^t, X_{n+1}|^{(t+1)} * |1 - d(\xi_{n+1}, x^{(i)})|$$

Avec :

$$d(\xi_{n+1}, x^{(i)}) = f^t(x^{(i)}) \cdot ((X_{n+1}^t, X_{n+1})^{(t+1)})^{-1} \cdot f(x^{(i)})$$

2.1. Algorithme de FEDOROV 1972

C'est un algorithme d'échange simultané qui garde toujours la taille n du plan désiré sans excursion. Après la génération d'un plan de départ aléatoire, l'algorithme sélectionne un point $x^{(i)}$ du plan et il doit être éliminé par un point $x^{(j)}$ de l'ensemble candidat. La procédure d'ajout et la suppression d'un point est faite en une seule étape et peut être démontrée par l'utilisation de la matrice d'information. En référence à l'équation (1) et (2), un échange est une addition et une soustraction simultanée d'un point.

Théorème 2 : Les matrices d'informations $(X_n^t \cdot X_n)^{(t+1)}$ et $(X_n^t \cdot X_n)^{(t)}$ sont liées par la relation :

$$(X_n^t \cdot X_n)^{(t+1)} = (X_n^t \cdot X_n)^{(t)} + f(x^{(j)}) \cdot f^t(x^{(j)}) - f(x^{(i)}) \cdot f^t(x^{(i)})$$

De même pour les déterminants :

$$|(X_n^t \cdot X_n)^{(t+1)}| = |(X_n^t \cdot X_n)^{(t)}| * \left| 1 + \Delta(\xi_n^{(t)}, x^{(j)}, x^{(i)}) \right| \quad (*)$$

Avec :

$$\Delta(\xi_n^{(t)}, x^{(j)}, x^{(i)}) = d(\xi_n^{(t)}, x^{(j)}) - \left(d(\xi_n^{(t)}, x^{(i)}) * (\xi_n^{(t)}, x^{(j)}) - d^2(\xi_n^{(t)}, x^{(j)}, x^{(i)}) \right) - d(\xi_n^{(t)}, x^{(i)})$$

$$d(\xi_n^{(t)}, x^{(j)}) = f^t(x^{(j)}) \cdot ((X_n^t \cdot X_n)^{(t)})^{-1} \cdot f(x^{(j)})$$

$$d(\xi_n^{(t)}, x^{(i)}) = f^t(x^{(i)}) \cdot ((X_n^t \cdot X_n)^{(t)})^{-1} \cdot f(x^{(i)})$$

$$d(\xi_n^{(t)}, x^{(j)}, x^{(i)}) = f^t(x^{(j)}) \cdot ((X_n^t \cdot X_n)^{(t)})^{-1} \cdot f(x^{(i)})$$

L'idée de base de l'algorithme de Fedorov est de calculer la valeur de Δ pour tous les couples possibles $(x^{(i)}, x^{(j)})$ et sélectionner l'une avec la plus grande valeur. Le point $x^{(i)}$ est pris à partir d'une sélection et $x^{(j)}$ peut être soit pris à partir des points restants qu'on appelle une recherche épuisante qui évite les points doubles dans le plan $n * (N - n)$ ou bien avec une recherche non épuisante, la sélection d'une expérience deux fois est possible $n * (N - 1)$. Comme l'équation (*) montre, les

points avec la plus haute Δ –augmentent la valeur du déterminant. De plus si plus un couple avec la même Δ est trouvé, l’algorithme choisit parmi eux d’une manière aléatoire.

Tandis que si un couple avec un effet Δ positif est trouvé, l’algorithme change les points et met à jour la matrice d’information et de dispersion. Parfois, on se trouve devant un couple qui augmente le déterminant mais pas d’une manière significative. Pour cela Fedorov a défini une valeur seuil ϵ_{fed} , si le maximum Δ est plus petite que ϵ_{fed} alors on arrête l’échange. En général on prend comme ϵ_{fed} la valeur 10^{-6} .

Algorithme

Générer un plan aléatoire avec un nombre de points désiré n ;

Tant que des couples avec Δ positive sont trouvés alors

Pour $i = 1$ à n faire

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour le point d’appui $x_{(i)}$ de plan;

Pour $j = 1$ à N faire ;

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour le point d’appui $x^{(j)}$;

Calculer la fonction de prédiction $d(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Calculer la fonction $\Delta(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Vérifier si le delta est maximum et enregistrer le couple ;

Si le maximum des deltas est positif alors

Si plus qu’un couple avec le même delta maximal alors

Sélectionner aléatoirement un couple ;

Changer le point $x_{(i)}$ sélectionné par $x^{(j)}$;

Mettre à jour la matrice d’information et la matrice de dispersion ;

Réinitialiser delta maximal ;

Algorithme de Fedorov

2.2. Algorithme de FEDOROV modifié

Cook et Nachtsheim 1980 font une comparaison de différents algorithmes pour construire des plans D-optimaux exacts et ils ont inventé un algorithme propre en fonction de l'algorithme de Fedorov. Ce dernier détermine la valeur de Δ pour tous les couples d'échanges possibles pendant une itération mais n'utilise qu'une seule valeur pour effectuer un échange. Ce calcul est coûteux lors de son utilisation. Avec la version modifiée par Cook et Nachtsheim, chaque itération de l'algorithme de Fedorov est décomposée en n étapes, une pour chaque point d'appui dans le plan au début de l'itération. Avec une matrice de conception d'un ordre aléatoire, l'algorithme commence avec le premier point d'appui $x^{(i)}$ et calcule les Δ pour tous les couples possibles avec ce point d'appui. Après avoir trouvé le meilleur échange de ce point, le plan est mis à jour et le point d'appui prochain est proposé pour un échange. Ce comportement est visualisé dans l'algorithme ci-dessous.

La différence entre les deux approches peut être montrée par l'exemple suivant : considérons un plan souhaité avec $n = 5$ points et un ensemble de candidats $N = 20$ expériences, $n * N = 100$ couples possibles peuvent être proposés pour un échange. A chaque itération de l'algorithme de Fedorov on calcule ces 100 valeurs Δ pour tous les couples possibles et on utilise une seule d'entre eux pour réaliser l'échange. En revanche, la version modifiée de l'algorithme commence par le premier point de plan et on ne calcule que 20 valeurs des Δ pour les couples possibles y compris ce point. Après cela, un échange est effectué et la matrice de dispersion doit être mise à jour. L'algorithme se poursuit avec le point du plan suivant et on calcule à nouveau les 20 valeurs du Δ . En général, à la fois l'algorithme calcule les 100 valeurs au cours d'une itération, la version d'échange calcule jusqu'à 5 points au cours de cette procédure.

Une étude réalisée par Cook et Nachtsheim montre que l'approche modifiée est peut être deux fois plus vite que l'algorithme de Fedorov.

Algorithme

Générer un plan aléatoire avec un nombre de points désiré n ;

Tant que des couples avec Δ positif sont trouvés **alors**

Pour $i = 1$ à n **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour le point d'appui $x_{(i)}$;

Pour $j = 1$ à N **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour le point $x^{(j)}$;

Calculer la fonction de prédiction $d(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Calculer la fonction $\Delta(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Sélectionner le couple avec un delta maximal ;

Si le maximum des deltas est positif **alors**

Si plus qu'un couple avec le même delta maximal **alors**

Sélectionner aléatoirement un couple ;

Changer le point $x_{(i)}$ sélectionné par $x^{(j)}$;

Mettre à jour la matrice d'information et la matrice de dispersion ;

Réinitialiser delta maximal ;

Algorithme de Fedorov modifiée

2.3. Algorithme de k-échange

Au cours d'une comparaison de l'algorithme de Fedorov et l'algorithme de Mitchell, Johnson et Nachtsheim 1983 ont montré que les points sélectionnés par l'algorithme de suppression de Federov ne sont normalement pas les uns avec la plus faible variance de prédiction, mais la fréquence des rangs de points supprimés pourrait être caractérisée comme biaisé en faveur des rangs de la variance inférieures. Simplement dit, au lieu de considérer tous les points candidats du plan ou seulement l'un avec la plus petite variance de prédiction, un ensemble de k points ayant la plus faible variance devait être sélectionnés. Similaire à l'algorithme de Federov modifié, une itération est maintenant décomposée en k étapes. A l'intérieur de chacune de ces k étapes les Δ -valeurs sont calculées et le couple correspondant serait échangé si le déterminant augmente.

La similitude avec les algorithmes précédemment décrits peuvent reconnue si nous définissons k -valeurs. Avec $k = 1$ l'algorithme est de Mitchell et $k = n$ il devient celui de Federov modifié. La sélection des k -valeurs est difficile et dans la plupart des cas dépendent de problème proposé. Une valeur commune qui est également suggéré par Johnson et Nachtsheim, est $k = 3$ ou $k = 4$. Meyer et Nachtsheim 1995 plus tard ont conseillé de sélectionner la valeur tel que :

$$k \leq \frac{n}{4}$$

Considérons l'exemple avec $n = 5$ et $N = 20$, le k -échange accélère le processus de sélection en réduisant le nombre compté pour le calcul de Δ . Similaire à la procédure de Fedorov modifiée, chaque itération de l'algorithme de base Fedorov est décomposé en Δ -calculs pour chaque point de plan. Mais en fonction de k -valeurs définis, l'algorithme tient compte que certains de plan. Avec $k = 3$ l'algorithme k -échnage calcule trois fois les 20 Δ - valeur et effectue jusqu'à trois échanges.

Cette approche a une plus grande efficacité de calcul par rapport à la procédure de Fedorov modifiée. En particulier pour les données de taille n grande, la définition de k -échange a un impact. La quantité du plan n'est pas toujours aussi bonne que celle de l'algorithme de Fedorov normale, mais en ce qui concerne le temps de calcul nécessaire, l'algorithme donne de bons résultats

Algorithme

Générer un plan aléatoire avec un nombre de points désiré n ;

Tant que des couples avec Δ positive sont trouvés **alors**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour tous les points de plan ;

Sélectionner k points de plan avec la variance de prédiction la plus faible ;

Pour $i = 1$ à k **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour le point $x_{(i)}$;

Pour $j = 1$ à N **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour le point $x^{(j)}$;

Calculer la fonction de prédiction $d(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Calculer la fonction $\Delta(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour le couple $(x_{(i)}, x^{(j)})$;

Sélectionner le couple avec un delta maximal ;

Si le maximum des deltas est positif **alors**

Si plus qu'un couple avec le même delta maximal **alors**

 Sélectionner aléatoirement un couple ;

Changer le point $x_{(i)}$ sélectionné par $x^{(j)}$;

Mettre à jour la matrice d'information et la matrice de dispersion ;

Réinitialiser delta maximal ;

Algorithme de k-échange

2.4. Algorithme kl-échange

C'est une modification de l'algorithme de Fedorov et donc un type du groupe 2 développé par Atkinson et Donev en 1989, cet algorithme permet de réduire la liste des points d'appuis et d'échanges avant de calculer la valeur Δ pour tous les couples possibles. L'utilisation du k points ayant la plus faible variance de prédiction est similaire à la procédure de k -échange. En plus de cela, nous ne considérons que les l candidats avec la plus grande variance de prédiction entre les points d'appuis.

En définissant $K = n$ et $l = N$, cet algorithme est la procédure de Fedorov. A l'initialisation normale de l'algorithme avec $k < n$ et $l < N$ réduit la qualité de calculer Δ et d'accélérer l'algorithme. Comme l'algorithme de Fedorov, le kl -échange

sélectionne les points dont la valeur Δ est la plus grande et effectue un seul échange, nous devons observer que les kl procédures s'appuient sur l'algorithme de Fedorov et n'effectue pas l'échange multiple au cours d'une itération. Seule la notion de choisir le point k de la plus faible variance de prédiction est similaire à l'algorithme de k -échange.

Atkinson et Donev 1989 ont défini deux modifications de cet algorithme : la première est semblable à l'idée de base de l'algorithme de Fedorov modifié par Cook et Nashtcheim et réalise plus d'un échange au cours de chaque itération. Avec la sélection de $k = 1$, cela à changer kl -échange et devient soit l'algorithme de Fedorov modifié avec $k = n$ ou k -échange avec $k < n$. La seconde modification modifie les critères de sélection pour les points k et l . l'algorithme sélectionne de façon aléatoire parmi les points de plan et les points d'appuis.

Dans l'algorithme de kl -échange, la version de base est montrée sans aucune modification. Par rapport à l'algorithme de Fedorov, le kl -échange réduit la quantité de calcul de Δ entre chaque itération. Pour notre exemple $n = 5, N = 20, k = l = 3$ conduit au calcul de seulement $k * l = 3 * 3 = 9$ valeurs de Δ . En comparaison avec les 100 couples de l'algorithme de Fedorov, cette approche kl accélère le calcul d'une manière efficace. Certes nous devons observer que la quantité d'itérations effectuée dans kl -échange peut être plus élevée par ce que tous les couples sont pris en considération pour un échange, mais en général l'algorithme kl -échange crée des modèles très rapides et dans la plupart des cas donnent des résultats acceptables compte tenu de la D-optimalité.

Algorithme

Générer un plan aléatoire avec un nombre de points désiré n ;

Tant que des couples avec Δ positif sont trouvés **alors**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour tous les points de plan ;

Sélectionner k points de plan avec la variance de prédiction la plus faible ;

Sélectionner l points de plan avec la variance de prédiction la plus grande ;

Pour $i = 1$ à k **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour le point $x_{(i)}$;

Pour $j = 1$ à l **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour le point d'appui $x^{(j)}$;

Calculer la fonction de prédiction $d(\xi_n^{(t)}, x^{(j)}, x_i)$ pour le couple $(x_{(i)}, x^{(j)})$;

Calculer la fonction $\Delta(\xi_n^{(t)}, x^{(j)}, x_i)$ pour le couple $(x_{(i)}, x^{(j)})$;

Vérifier si le delta est le maximum et enregistrer le couple ;

Si le maximum des deltas est positif **alors**

Si plus qu'un couple avec le même delta maximal **alors**

Sélectionner aléatoirement un couple ;

Changer le point $x_{(i)}$ sélectionné par $x^{(j)}$;

Mettre à jour la matrice d'information et la matrice de dispersion ;

Réinitialiser delta maximal ;

Algorithme kl-échange

2.5. Algorithme kl-échange modifié

Une modification de l'algorithme kl-échange est développée. Le changement dans l'algorithme est d'étendre la valeur k si les k points sélectionnés ont la même variance de prédiction et d'un analogue ($k + 1^{ième}$) valeur existe. En utilisant une sélection aléatoire si plus de k candidats ont la même variance, alors elle peut aussi être une solution, mais elle n'est pas aussi infiable. La deuxième modification tente d'empêcher l'algorithme kl-échange de l'échange de certains couple qui ne sont pas les meilleurs choix en réduisant la liste des points d'appuis. Commenant par $l=N$ points dans l'ensemble de points d'appuis possibles, l'algorithme supprime les points en fonction de critères différents. En règle générale, un échange ne doit être effectué

si la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ du point de support $x^{(j)}$ est supérieure à la variance de prédiction $d(\xi_n^{(t)}, x^{(i)})$ du point de conception $x^{(i)}$ supprimé. Un deuxième critère de réduire la liste des points de support est développé dépend de la variance moyenne de la prédiction à partir de la dernière itération .seuls les points d'appuis avec une variance plus élevée de prédiction que la variance moyenne est prise en compte. Dans ce cas, la variance moyenne de prédiction est calculée avec l'emploi de tous les candidats possibles, le support restant des points est vérifié.

Cet algorithme trouve des difficultés pour comparer les approches précédentes parce que le montant du point de conception et de soutien compte pour un échange est dépendant de problème. Nous ne faisons pas valeurs pour l et k et les critères pour la sélection des points peuvent changer au cours du processus, mais en général, la modification kl -échange est comparable avec kl -échange normal. Les calculs doivent un peu plus de temps parce que nous changeons parfois les critères de la variance et on calcule le meilleur échange. Mais dans l'ensemble, l'algorithme est comparable au kl -échange normal. Compte tenu de la qualité des motifs générés, cet algorithme conduit aux meilleurs résultats dans la plupart des cas, vaut également la procédure k -échange.

Bien sûr, les résultats n'ont pas la même qualité que les dessins créés avec l'algorithme de Fedorov, mais avec un temps de calcul recevable, nous créons une bonne D-optimalité.

Algorithme

Générer un plan aléatoire avec un nombre de points désiré n ;

Tant que des couples avec Δ positive sont trouvés **alors**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour tous les points du plan ;

Si plus de k points de plan ont la même faible variance de prédiction **alors**

étand donné k ;

Sélectionner k points de conception avec la plus faible variance de prédiction ;

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour tous les points d'appuis ;

Calculer la variance moyenne de prédiction sur l'ensemble des points candidats ;

Sélectionner l points d'appuis avec une plus grande variance de prédiction que la moyenne valeur ;

Pour $i = 1$ à k **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour ce point de conception ;

Pour $j = 1$ à l **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour ce point d'appui ;

Si la variance du point d'appui est supérieure à la variance du point de conception **alors**

calculer la fonction de prédiction $d(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour ce couple ;

Si delta maximal est négatif **alors**

Sélectionner $N - l$ points d'appui qui n'ont pas été vérifiés ;

Pour $i = 1$ à k **faire**

calculer la variance de prédiction $d(\xi_n^{(t)}, x_{(i)})$ pour ce point de conception;

Pour $j = 1$ à $N - l$ **faire**

Calculer la variance de prédiction $d(\xi_n^{(t)}, x^{(j)})$ pour ce point d'appui ;

Calculer la fonction de variance $d(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour ce couple ;

Calculer la fonction delta $\Delta(\xi_n^{(t)}, x^{(j)}, x_{(i)})$ pour ce couple ;

Si delta maximal est positif **alors**

Si plus qu'un couple avec le même delta maximal **alors**

Sélectionner aléatoirement un couple ;

Changer le point $x_{(i)}$ sélectionné par $x^{(j)}$;

Mettre à jour la matrice d'information et la matrice de dispersion ;

Réinitialiser delta maximal ;

Algorithme de kl-échange modifié

2.6. Algorithme FDOP (Yonchev)

Suivant l'échange proposé par FEDOROV on remarque que le point choisi n'est pas toujours le point ayant la fonction de variance la plus élevée car il cherche le meilleur couple. Pour cette raison Yonchev évite la première étape en ajoutant aléatoirement M points candidats pour en extraire par la suite successivement les M mauvais points.

* $\xi_n^{(t)}$ (ajout de M points) $\rightarrow \xi_{n+M}^{(t+1)}$.

* $\xi_{n+M}^{(t+1)}$ (Suppression de M points donnant la diminution minimale) $\rightarrow \xi_n^{(t+2)}$.

Le critère d'arrêt est le nombre maximal d'excursions (*maxi*) n'ayant pas modifié la structure de ξ_n . L'expérience montre que prendre $maxi \leq 4$ suffit pour obtenir des résultats comparables à ceux obtenus par les algorithmes de Fedorov. L'amélioration des temps de calcul dépend du choix de M . Plus M est petit, plus le traitement d'un lot est rapide, mais plus il y a des lots à traiter. L'étude de l'influence de M montre que les meilleurs résultats sont obtenus pour des valeurs de M comprises dans l'intervalle $[p/2, 3p]$ ou p est le nombre de paramètre du modèle. Pour cet algorithme, nous notons $X_k^{(i)j}$: la matrice du modèle comportant k expériences testées durant l'excursion longue i et l'excursion courte j .

Algorithme

Initialiser la matrice de départ $X_n^{(0)0}$

Tant que (nombre d'excursion $<$) **alors**

Initialiser des lots

Crée R lots contenant aléatoirement M points de la liste des expériences candidates.

Exécution courte

Pour $j = 1$ à R **faire**

Calculer le déterminant et la matrice de dispersion de $X_n^{(t)0}$

Ajouter les M points du $j^{\text{ème}}$ lot $\rightarrow X_{n+M}^{(t)j}$

Caractériser la nouvelle matrice

Calculer le déterminant et la matrice de dispersion

Éliminer les M points mauvais

Tant que (nombre de points dans la matrice $>$) **alors**

Calculer les variances de prédiction

Le point dont la variance de prédiction est minimale est retiré de la matrice.

Validation d'une excursion courte

Si ($|(X_n^t \cdot X_n)^{(t)0}| < |(X_n^t \cdot X_n)^{(t)j}|$) **alors**

$X_n^{(t)0} = X_n^{(t)j}$

Validation d'une excursion

Si ($|(X_n^t \cdot X_n)^{(t-1)0}| < |(X_n^t \cdot X_n)^{(t)0}|$) **alors**

Incrémenter le nombre d'excursions

Sinon

Le nombre d'excursions = 0

Algorithme de FDOP

3. APPLICATION DES ALGORITHMES D'ÉCHANGES SUR LA FABRICATION PHARMACEUTIQUE D'UN MÉDICAMENT BIEN PROTÉGÉ

3.1. Préparation de l'expérimentation

Une société pharmaceutique fabrique un médicament très actif, mais malheureusement sensible à la température et à l'humidité. Pour augmenter la durée de conservation, on ajoute un desséchant au conditionnement du médicament. Un technicien est chargé d'optimiser la quantité de desséchant qu'il faut introduire. Le matériel dont il dispose pour réaliser son étude lui impose des contraintes. Toutes les combinaisons de niveaux ne sont pas possibles. Il ne peut pas, non plus, faire trop d'essais car la fabrication attend des résultats pour démarrer. Pour surmonter toutes ces contraintes, il choisit de conduire ses expériences selon un plan D-optimal.

3.2. Description des contraintes

Le médicament à étudier se décompose et perd son activité à la fois sous l'influence de la chaleur mais aussi sous l'influence de l'humidité. Ces deux facteurs ont des actions néfastes qui parfois se renforcent et qui parfois s'annihilent l'une l'autre suivant les conditions de conservation. Le médicament doit être étudié entre 30° à 50° pour la température et entre 50% et 90% pour l'humidité. Le technicien possède deux étuves pour réaliser ces essais. La première étuve couvre les températures de 30° à 40° et les humidités de 50% et 90%. Le second couvre les températures de 40° à 50° et les humidités de 50% à 80%. Il n'est pas possible d'atteindre 90% d'humidité avec cette dernière étuve.

Le médicament se décompose en une substance facile à déterminer et à doser. Cette substance n'est pas toxique, mais elle est inerte. La disparition du médicament principal diminue l'activité thérapeutique du produit vendu et c'est ce qu'il faut éviter.

3.3. Choix des facteurs

Le technicien retient 3 facteurs

1^{ère} : La température en °C

2^{ème} : Le pourcentage d'humidité

3^{ème} : La masse du desséchant en *g*

3.4. Domaine d'étude

Le tableau suivant indique les niveaux hauts et bas de chaque facteur. On sait que le point 50° et 90% d'humidité ne peut pas être réalisé avec le matériel disponible.

Facteur	Niveau -1	Niveau 0	Niveau +1
Température (1)	30°C	40°C	50°C
Humidité (2)	15%	70%	90%
Desséchant (3)	1g	2g	3g

Tableau 6 : Un médicament bien protégé. Domaine d'étude.

3.5. Choix des réponses

La réponse choisie par le technicien est la masse de la substance issue de la décomposition du médicament. Cette substance est facile à isoler et peser avec une bonne précision.

Etant donné la complexité des processus de dégradation du médicament, les modèles de réponses ne sont pas probablement de premier degré. Il faut envisager des modèles du second degré.

3.6. Choix du plan expérimental

Le domaine d'étude est tronqué puisqu'il n'est pas possible de réaliser tous les points d'expériences. Le technicien décide de couvrir le domaine accessible avec un grand nombre de points d'expériences et d'utiliser l'algorithme D-optimal pour choisir les plus pertinents.

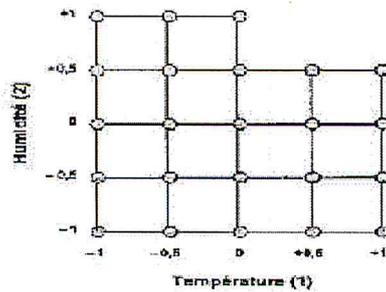


Figure 3 : Disposition des 23 points d'expériences candidats pour un niveau du facteur 3.

Comme il y a trois niveaux pour ce facteur, il y a 69 points candidats.

Le technicien attribue trois niveaux au facteur 3 (masse de desséchant) sur le quel ne pèse aucune contrainte expérimentale. Les facteurs 1 et 2 auront les niveaux -1 , -0.5 , 0 , +0.5 , +1 à chaque fois que cela sera possible. Pour un niveau du facteur 3, les points d'expériences candidats seront disposés comme l'indique la figure précédente. Il y a 23 points d'expériences candidats par niveau du facteur 3.

Il y a donc tout $N = 69$ essais possibles. Mais, comme le travail doit être réalisé le plus vite possible, le technicien fait tourner l'algorithme de calcul des plans D-optimaux afin de choisir les expériences les plus favorables à l'établissement du modèle du second degré. Ce modèle est le modèle complet qui possède dix coefficients. Le technicien demande douze points d'expériences donc = 12 .

Pour déterminer la matrice d'expérience X , on va utiliser les algorithmes précédents et puis on calcule le déterminant de $(X^t X)$ dont les résultats sont dans le tableau.

ALGORITHMES	$\det(X^t X)$
FEDOROV	8.9471×10^6
FEDOROV MODIFIE	1.1711×10^7
K-ECHANGE	5.1796×10^6
KL-ECHANGE	10.5469
FDOP	1.2224×10^4

Tableau 7 : Le déterminant de $(X^t X)$ suivant les algorithmes

Dans cet exemple on remarque que par l'algorithme de Fedorov modifié on trouve le déterminant très élevé d'où l'optimalité.

4. COMPARAISON ENTRE LES ALGORITHMES D'ÉCHANGE

En général, une comparaison des algorithmes qui doit être fait en fonction de deux critères principaux. Le temps de calcul utilisé pour créer un plan D-optimal, et le second est la qualité de la conception. Dans cette analyse on va comparer les algorithmes suivant l'application précédente, nous utilisons la D-optimalité où son analogue, le facteur déterminant pour évaluer les plans optimaux.

La base des données fournies dans ce travail est une analyse des six algorithmes mise en œuvre avec MATLAB. Pendant le déroulement du programme pour calculer le déterminant et le temps de calcul est mesuré, les données recueillies peuvent être trouvées dans le tableau.

ALGORITHMES	$\det (X^t X)$	Temps d'exécution / s
FEDOROV	8.9471×10^6	0.126250
FEDOROV MODIFIE	1.1711×10^7	0.040134
K-ECHANGE	5.1796×10^6	0.022058
KL-ECHANGE	10.5469	0.24881
FDOP	1.2224×10^4	0.938205

Tableau 8 : Comparaison des algorithmes d'échanges.

L'algorithme de Fedorov, par exemple, calcule la valeur pour tous les couples possibles et effectue dix échanges en fonction de ces données, avec un grand nombre de facteurs, tandis que la modification qui était faite par Cook & Nachtsheim (1980) effectue quinze échanges. Le temps de calcul moyen utilisé pour l'algorithme Fedorov modifié peut être rapide jusqu'à 75% que l'algorithme normal, surtout avec beaucoup de données, la différence entre les deux algorithmes devient évidente.

Algorithmes de k-échange par Johnson & Nachtsheim (1983) est beaucoup plus rapide dans la création des plans D-optimaux et peut être utilisé pour un très grand nombre de facteurs. Il est deux fois plus rapide que la procédure de Fedorov modifié.

Le Fedorov modifié et Fedorov créent normalement les meilleurs plans, mais en général, la sélection du meilleur algorithme est dépendante de problème. Par contre, l'algorithme de k-échange est meilleur que FEDOROV, kl-échange et Fdop par rapport au temps d'exécution et si on compare par rapport au déterminant on trouve qu'il est aussi meilleur que celui de Fdop et le kl-échange.

Chapitre 3

CONSTRUCTION ALGORITHMIQUE DES PLANS D-OPTIMAUX PAR DEPLACEMENT

1. LES ALGORITHMES DE DEPLACEMENT

Avec le temps, de nouvelles méthodes sont découvertes pour construire les plans D-optimaux après les algorithmes d'échanges dont le principe est d'utiliser les matrices proches de l'optimalité (c.à.d. une matrice déjà générée par un algorithme d'échange) comme une solution de départ (réalisable). Dans ce chapitre on va présenter trois méthodes qui sont les plus utilisables.

1.1. Algorithme de DONEV (Donev et Atkinson)

Cet algorithme propose un moyen d'améliorer des matrices quasi optimales, obtenues par exemple par les algorithmes d'échanges. Il explore les alentours de chacun des points de la matrice, afin d'obtenir une solution plus performante en occupant des positions plus précises.

A partir d'une matrice proche de l'optimalité ils réalisent une amélioration de la position des points sans bouleverser la configuration. Ils effectuent sur chacun des points de la matrice des variations pour déterminer lesquels des n points doivent être modifiés et dans quelle direction. Le principe de cet algorithme consiste à évaluer la variation du déterminant de la matrice d'information en faisant varier une coordonnée d'un point de la matrice, ce qui revient à perturber la matrice d'expérience en modifiant une valeur et en évaluant les effets sur le déterminant de la matrice d'information[5].

Exemple : Dans le cas où notre domaine est défini par deux variables indépendantes, soit x_i le point qui nous désirons faire varier. Nous testerons les quatre points A, B, C et D tels qu'ils sont représentés sur le schéma. Le point x_i est déplacé sur la position qui donne le plus grand déterminant de $(X^t X)$.

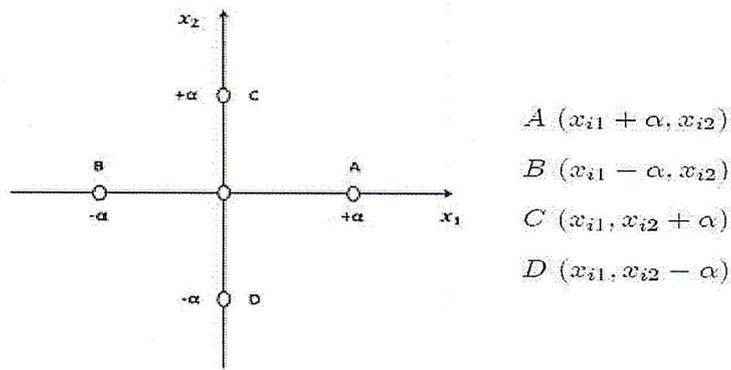


Figure 4 : Déplacements de Donev en 2 dimensions.

L'algorithme calcule ainsi $(2.k)$ déplacements et donc $(2k)$ fonctions «delta» par étape. Nous pouvons encore réduire le temps de calcul en utilisant une méthode rapide pour recalculer les termes de $(X^t X)^{-1}$ après le déplacement.

L'algorithme proposé par Donev et Atkinson, très performant sur des matrices proches de l'optimalité-D, est le suivant :

Algorithme

```

Choix d'une amplitude de déplacement  $\alpha$ 
Calculer la matrice  $(X^t X)$ 
Tant que  $(\alpha \geq \alpha_{min})$  alors
   $\Delta_{max} = 0$ 
  Essayer pour tous les points
  Pour  $i = 1$  à  $n$  faire
    Essayer tous les changements sur les axes
    Pour  $j = 1$  à nombre de variables indépendantes faire
      Essayer sens négatif puis positif
      Pour sens=-1 puis +1 faire
         $D = (d_1, d_2, \dots, d_n)$  avec  $d_j = \text{sens}$  et  $d_m = 0$  (pour  $m \neq j$ )
        Calculer le nouveau point  $x' = x_i + \alpha \cdot D$ 
        Calculer la fonction delta de Fedorov :  $\Delta$ 
        Si  $(\Delta > \Delta_{max})$  alors
           $x'$  Est le changement sélectionné
           $\Delta_{max} = \Delta$ 
      Changement ou convergence
    Si  $\Delta_{max} > 1$  alors
      Effectuer le meilleur changement sélectionné
      Calculer de la nouvelle matrice  $(X^t X)^{-1}$ 
  Sinon
     $\alpha = \alpha/2$ 

```

Algorithme de Donev.

1.2. Algorithme de FDOP modifié

Cette méthode est basée sur le principe de Yonchev et inspirée des méthodes à déplacement aléatoires. Elle provoque des perturbations sur une partie des points pour détecter si une modification aléatoire peut améliorer la solution [5].

Algorithme

```
Initialiser la matrice de départ  $X_n^{(1)0}$ 
Tant que (nombre d'excursions < 4) alors
  Crée  $R$  lots contenant  $M$  numéros de points.
  Excursion courte
  Pour  $j = 1$  à  $R$  lots faire
    Matrice de dispersion de la matrice actuelle  $\rightarrow X_n^{(1)0}$ 
    Générer les  $M$  nouveaux points
    Pour  $i = 1$  à  $M$  faire
       $l =$  numéro du  $i^{\text{ème}}$  point dans le  $j^{\text{ème}}$  lot
       $D =$  choix du déplacement unitaire est aléatoire
       $x_{n+l} = x_l + \alpha \cdot D$ 
    Eliminer les  $M$  mauvais points
    Tant que  $i \neq 0$  alors
      Matrice de dispersion de la nouvelle matrice  $X_{n+i}^{(t)j}$ 
      Calculer la variance de prédiction  $d(x_{n+i})$ 
      Eliminer le point ayant la variance de prédiction minimale
       $i = i - 1$ 
    Validation d'une excursion courte
  Si ( $|(X_n^t \cdot X)^{(t)0}| < |(X_n^t \cdot X)^{(t)j}|$ ) alors
     $X_n^{(t)0} = X_n^{(t)j}$ 
  Validation d'une excursion
  Si ( $|(X_n^t \cdot X)^{(t-1)0}| < |(X_n^t \cdot X)^{(t)0}|$ ) alors
    Incréments nombre d'excursions
  Sinon
    nombre d'excursion = 0
```

Algorithme de FDOP modifié

2. APPLICATION DES ALGORITHMES DE DEPLACEMENT SUR L'ETUDE EXPLOSIVE

Un explosif ne doit pas présenter de risque d'explosion en cas de choc accidentel. Il doit pouvoir être transporté et conservé en toute sécurité. Mais les moyens mis en œuvre pour le rendre sûr ne doivent pas diminuer sa puissance explosive. Le formulateur doit trouver le meilleur compromis entre sécurité et efficacité. C'est l'un de ses soucis majeurs. Les compositions explosives ont été classées en plusieurs catégories dont la plus recherchée est MDEPS (matière détonante extrêmement peu sensible). De telles compositions sont obtenues en associant judicieusement des composés actifs. L'objectif de l'étude est de trouver une composition moins sensible que la composition H6 réputée trop sensible. Une composition est sensible si elle présente des risques élevés d'explosion spontanée lorsqu'elle est soumise à des agressions telles qu'une variation de température ou un léger choc.

Les essais sur les matières explosives demandent une préparation précise et une organisation sans faille. Les compositions sont délicates à réaliser et nécessitent des protocoles de fabrication longs et rigoureux. Les campagnes d'évaluation des propriétés explosives et les mesures mettent en œuvre d'importants moyens. Le nombre des expériences doit être limité au strict nécessaire. L'économie d'un seul essai entraîne un gain de temps et une économie appréciable. C'est l'une des situations où les plans D-optimaux sont précieux.

2.1. Description de la composition explosive

La composition H6 contient quatre constituants

- L'aluminium,
- RDX, explosif très réactif,
- La cire,
- Un explosif fusible, le TNT (trinitrotoluène).

Cette composition est satisfaisante comme explosif, mais elle ne passe pas les tests permettant de la classer en MDEPS. Pour diminuer la sensibilité du produit, les expérimentateurs explorent plusieurs voies :

- Un meilleur équilibre entre les proportions des trois premiers constituants, l'aluminium, l'explosif RDX et la cire,
- Le remplacement du RDX, explosif réputé trop sensible, par le couple HMX/ONTA qui l'est beaucoup moins,
- L'utilisation d'un nouvel explosif fusible, le TNMA à la place du TNT.

2.2. Choix des facteurs

Les pyrotechniciens retiennent quatre facteurs :

Facteur 1 : la proportion d'aluminium.

Facteur 2 : le rapport HMX sur ONTA,

Facteur 3 : la proportion de cire.

Facteur 4 : le type d'explosif fusible.

2.3. Domaine d'étude

Le tableau suivant indique les niveaux hauts et bas de chaque facteur

Facteur	Niveau -1	Niveau +1
Aluminium (1)	8%	20%
Rapport HMX sur ONTA (2)	0/100	50/50
Cire (3)	4%	10%
Explosif fusible (4)	TNMA	TNT

Tableau 9 : Une étude explosive. Domaine d'étude.

2.4. Choix des réponses

Deux réponses sont choisies par les responsables de l'étude :

- La vitesse de détonation en mètres par seconde. Cette grandeur caractérise la force explosive du produit. On vise une vitesse de détonation d'au moins 7400m/s,
- Le coefficient de sensibilité qui caractérise la susceptibilité du produit aux sollicitations extérieures. Cette sensibilité doit être faible, inférieure à 95 si possible.

2.5. Choix du plan expérimental

Il y a trois facteurs continus et un facteur discontinu. Les expérimentateurs pensent que les réponses suivent une loi du second degré pour les facteurs continus. Il faut donc choisir un plan permettant d'obtenir ce modèle. Un plan de Doehlert leur semble approprié. Treize essais doivent être envisagés pour ces trois facteurs.

Quant au facteur discontinu (type d'explosif fusible), il n'a que deux niveaux.

Le plan complet entraîne la réalisation de 26 essais. C'est beaucoup trop. Les expérimentateurs décident donc de choisir un petit nombre d'essais parmi les 26 possibles. Ce choix est réalisé grâce au critère de D-optimalité, qui oblige à choisir, avant les essais, un modèle mathématique.

2.6. Modèle postulé

Les expérimentateurs adoptent un modèle du second degré pour les trois facteurs continus. Ils n'envisagent pas d'interaction entre ces facteurs. Le modèle possède donc un terme constant, des termes du premier degré et des termes du second degré. Il n'y a pas de termes rectangles.

Le facteur discontinu ne prend que deux niveaux. On le modélise avec un terme du premier degré et on ne prévoit pas d'interaction entre ce facteur et les trois autres.

Le modèle à priori est le même pour les deux réponses, vitesse de détonation et sensibilité. Le modèle est le suivant :

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 + e$$

Il y a huit coefficients à déterminer. Si l'on veut ménager quelques degrés de liberté pour évaluer le résidu, on réalisera entre 10 et 12 expériences, soit moins de la moitié du plan prévu initialement.

Essai n°	Aluminium (1)	HMX sur ONTA (2)	Cire (3)	Explosif fusible (4)
1	0	0	0	-1
2	+1	0	0	-1
3	+0.5	+0.866	0	-1
4	-0.5	+0.866	0	-1
5	-1	0	0	-1
6	-0.5	-0.866	0	-1
7	+0.5	-0.866	0	-1
8	+0.5	+0.289	+0.816	-1
9	-0.5	+0.289	+0.816	-1
10	0	+0.577	+0.866	-1
11	+0.5	-0.289	-0.816	-1
12	-0.5	-0.289	-0.816	-1
13	0	+0.577	-0.816	-1
14	0	0	0	+1
15	+1	0	0	+1
16	+0.5	+0.866	0	+1
17	-0.5	+0.866	0	+1
18	-1	0	0	+1
19	-0.5	-0.866	0	+1
20	+0.5	-0.866	0	+1
21	+0.5	+0.289	+0.816	+1
22	-0.5	+0.289	+0.816	+1
23	0	-0.577	+0.816	+1
24	+0.5	-0.289	-0.816	+1
25	-0.5	-0.289	-0.816	+1
26	0	+0.577	-0.816	+1

Tableau 10 : Une étude explosive. Points candidats du projet initial.

2.7. Calcul du plan D-optimal

Les expérimentateurs possèdent un logiciel de plan d'expériences pourvu de l'algorithme de calcul des plans D-optimaux. Ils commencent par entrer les 26 points candidats indiqués par le tableau précédent. Ils indiquent ensuite au logiciel, le modèle qu'ils ont retenu et ils lancent l'algorithme de calcul en imposant un nombre de $n = 12$ expériences. Après plusieurs passages, ils retiennent les expériences données par la machine et écrivent le tableau ci-dessous. C'est le plan D-optimal qui va être réalisé. Ces douze essais permettant de déterminer les huit coefficients du modèle, on va utiliser les algorithmes précédents et puis on calcule le déterminant de $(X'X)$ dont les résultats sont dans le tableau suivant.

ALGORITHMES	$\det(X'X)$
DONEV	7.4139×10^3
FDOP MODIFIE	1.2757×10^4

Tableau 11 : Le déterminant de $(X'X)$ suivant les algorithmes.

3. COMPARAISON DES ALGORITHMES

Les algorithmes qui sont traités dans ce chapitre obligent d'effectuer plusieurs fois la même recherche pour augmenter la probabilité d'obtenir la solution idéale. Nous effectuons donc pour toutes les méthodes 10 fois la même recherche et comparons les meilleures solutions obtenues.

En pratique, il est impossible de considérer l'infinité des niveaux donc dans tous les cas nous accepterons une incertitude relative sur la position d'un point expérimental de l'ordre de 0,005.

Le tableau suivant montre les différents déterminants calculés par les algorithmes ainsi que leurs temps d'exécutions.

ALGORITHMES	$\det (X'X)$	Temps d'exécution /s
DONEV	7.4139×10^3	0.049183
FDOP MODIFE	1.2757×10^4	0.191576

Tableau 12 : Comparaison des algorithmes.

Dans cet exemple on remarque que par l'algorithme de Fdop modifié on trouve le déterminant le plus élevé d'où l'optimalité. Par contre l'algorithme de Donev est meilleur que Fdop modifié par rapport au temps d'exécution.

Chapitre 4

LES NOUVELLES PROPOSITIONS

1. PRINCIPE DE L'ALGORITHME

DONEV modifié propose un moyen d'améliorer l'algorithme de Donev à partir d'une matrice presque optimale où il fait des déplacements. Cet algorithme déplace le premier point de la matrice après avoir choisi le meilleur point du plan tout en utilisant la fonction de Fedorov Δ et on répète cette procédure plusieurs fois et à la fin on fait l'échange. Puis on répète la même procédure pour le point suivant, à la fin on obtient une matrice d'expérience dont le déterminant est meilleur que le précédent d'où l'optimalité.

2. ALGORITHME PROPOSE

Pour l'algorithme de Donev on déplace tous les points expérimentaux point par point tout en sélectionnant le meilleur point, ensuite on fait l'échange si la valeur de la fonction de Fédérov est supérieure ou égale à 1, si non on divise la valeur de paramètre α par deux et dans les deux cas on répète la même procédure jusqu'à atteindre α_{min} . Tandis que, dans l'algorithme que l'on propose, on déplace le premier point et si la valeur de la fonction de Fédérov du meilleur déplacement est inférieur à 1 on divise le paramètre α par deux, si non on garde l'ancienne valeur de α , puis on répète cette même procédure au niveau de ce meilleur point déplacé jusqu'à atteindre la valeur de α_{min} . S'il n'y a aucune amélioration de la valeur de Fédérov après cette opération de déplacement concernant le premier point, on fait l'échange du premier point avec le meilleur déplacement trouvé puis on passe au deuxième point expérimental jusqu'au dernier point.

Algorithme

Choix d'une amplitude de déplacement α

Calculer la matrice $(X'X)$

Pour $i = 1$ à n **faire**

$t = i$

$s = i$

$\Delta_{max} = 0$

 Initialiser α

Tant que $(\alpha > \alpha_{min})$ et (le point x_t est amélioré) **alors**

 Essayer tous les changements sur les axes

Pour $j = 1$ à nombre de variables indépendantes **faire**

 Essayer sens négatif puis positif

Pour sens -1 puis $+1$ **faire**

$D = (d_1, d_2, \dots, d_n)$ avec $d_j = \text{sens}$ et $d_m = 0$ (pour $m \neq j$)

 Calculer le nouveau point $x_q = x_t + \alpha \cdot D$

 Si x_q dans le domaine **alors**

 Calculer la fonction delta de Fedorov : $\Delta(i, q)$

Si $(\Delta(i, q) > \Delta_{max})$ **alors**

x_q est le changement sélectionné

$\Delta_{max} = \Delta(i, q)$

$s = q$

Si $(\Delta_{max} > 1)$ **alors**

$\alpha = \alpha$

Sinon

$\alpha = \alpha/2$

$t = s$

Changement ou convergence

Si $(\Delta_{max} > 1)$ **alors**

 Effectuez le meilleur changement sélectionné c'est-à-dire on change x_i par x_q .

 Calculez la nouvelle matrice $(X'X)^{-1}$

Algorithme de Donev modifié.

3. ETUDE COMPARATIVE

Les expérimentations ont été réalisées sur un PC équipé d'un processeur Intel(R) Core (TM) i5 -3230M CPU @ 2.60 GHz avec une mémoire de 4 GB sous Windows 7 Professionnel. Les algorithmes sont implémentés en MATLAB. Les programmes prennent comme donnée d'entrée des matrices d'expériences contenant des points et doivent être changées à la fin de l'algorithme pour arriver à l'optimalité.

Deux critères de comparaisons sont utilisés : la valeur du déterminant et le temps d'exécution.

Pour le cas de deux et trois facteurs et un modèle de premier et troisième degré, l'algorithme proposé donne un meilleur résultat par rapport au temps d'exécution et avec le même déterminant que celui trouvé par DONEV et celui de FDOP. Et toujours pour deux facteurs mais pour un modèle de deuxième degré on a constaté que les deux algorithmes DONEV et l'algorithme proposé donnent le même déterminant mais en temps d'exécution l'algorithme proposé est le meilleur sauf pour α égale à 0.1 où on a trouvé que le temps d'exécution de DONEV est plus rapide que celui de l'algorithme proposé .

Tant dis que pour d'autres exemples choisis aléatoirement on a trouvé que pour les mêmes valeurs de α toujours l'algorithme proposé donne le meilleur résultat en déterminant par rapport à celui donné par DONEV mais pas en temps d'exécution sauf pour α égale à 0.1, les trois algorithmes donnent le même déterminant mais en temps d'exécution notre algorithme proposé est toujours le meilleur.

Pour trois facteurs et pour un modèle de deuxième degré, l'algorithme proposé donne un meilleur résultat en déterminant même en temps. Par contre pour un modèle de troisième degrés le temps d'exécution de notre algorithme est le meilleur pour toutes les valeurs de paramètre α et les trois algorithmes comparés donnent le même déterminant.

Pour quatre facteurs et pour un modèle de premier degré on a trouvé que les trois algorithmes donnent le même déterminant mais en temps d'exécution notre algorithme est le meilleur que celui de DONEV et FDOP sauf pour α égale à 1.5. Par contre pour un modèle de troisième degré notre algorithme est le meilleur en déterminant et

en temps d'exécution et pour un modèle de deuxième degré les résultats trouvés sont différents selon les valeurs de α .

Pour cinq, six et sept facteurs nous avons trouvé des résultats remarquables pour l'amélioration de déterminant et du temps d'exécution et cela pour différents type de modèle choisis à priori. Pour cinq facteurs nous avons obtenu une amélioration en temps et en déterminant et cela pour les deux modèles choisis. Pour six facteurs et avec un modèle de premier degré nous avons amélioré le temps et le déterminant, par contre pour un modèle de troisième degré les résultats sont différents selon les valeurs de paramètre α . Pour sept facteurs, l'amélioration à été en même temps en temps d'exécution et en déterminant.

Pour tous les exemples que nous avons traités, nous avons trouvé que notre algorithme a donné toujours des meilleurs résultats par rapport à l'algorithme de FDOP.

Ces résultats sont regroupés dans les tableaux ci-dessous.

Cas d'un plan avec 2 facteurs : $k = 2$ $N = 9$ $n = 3$ $p = 3$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	16	16	16
	Temps	0.119172	0.126406	3.644074
0.5	déterminant	16	16	16
	Temps	0.122177	0.094426	3.623594
1	déterminant	16	16	16
	Temps	0.144489	0.116944	3.597560
1.5	Déterminant	16	16	16
	Temps	0.123447	0.099014	3.284485

Tableau 13 : plan avec 2 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 2 facteurs : $k = 2$ $N = 24$ $n = 11$ $p = 6$

Modèle de 2^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$
	temps	0.519357	0.559448	1.972052
0.5	déterminant	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$
	temps	0.512548	0.422320	1.854122
1	déterminant	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$
	temps	0.560694	0.529041	1.885231
1.5	déterminant	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$	$1.1168 * 10^{14}$
	temps	0.577599	0.512866	2.005143

Tableau 14 : plan avec 2 facteurs, modèle de 2^{ème} degré

Cas d'un plan avec 2 facteurs : $k = 2$ $N = 24$ $n = 11$ $p = 6$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$1.5522 * 10^8$	$1.5522 * 10^8$	$1.5522 * 10^8$
	temps	0.480345	0.417185	1.533270
0.5	déterminant	$1.5522 * 10^8$	$1.5522 * 10^8$	$1.5522 * 10^8$
	temps	0.491430	0.449868	1.639168
1	déterminant	$1.5522 * 10^8$	$1.5522 * 10^8$	$1.5522 * 10^8$
	temps	0.482362	0.361121	1.405013
1.5	déterminant	$1.5522 * 10^8$	$1.5522 * 10^8$	$1.5522 * 10^8$
	temps	0.457439	0.450563	1.469608

Tableau 15 : plan avec 2 facteurs, modèle de 3^{ème} degré

Cas d'un plan avec 3 facteurs : $k = 3$ $N = 69$ $n = 12$ $p = 10$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$1.6164 * 10^4$	$1.6164 * 10^4$	$1.6164 * 10^4$
	temps	0.660081	0.607922	1.984169
0.5	déterminant	$1.6164 * 10^4$	$1.6164 * 10^4$	$1.6164 * 10^4$
	temps	0.633265	0.606274	1.551307
1	déterminant	$1.6164 * 10^4$	$1.6164 * 10^4$	$1.6164 * 10^4$
	temps	0.650276	0.615359	1.394561
1.5	déterminant	$1.6164 * 10^4$	$1.6164 * 10^4$	$1.6164 * 10^4$
	temps	0.661165	0.620545	1.303127

Tableau 16 : plan avec 3 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 3 facteurs : $k = 3$ $N = 27$ $n = 12$ $p = 10$

Modèle de 2^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$7.6639 * 10^{16}$	$7.6639 * 10^{16}$	$7.6639 * 10^{16}$
	temps	0.586364	1.009785	2.248308
0.5	déterminant	$7.6639 * 10^{16}$	$1.5602 * 10^{17}$	$7.6639 * 10^{16}$
	temps	0.664403	0.475684	2.297550
1	déterminant	$7.6639 * 10^{16}$	$1.5427 * 10^{17}$	$7.6639 * 10^{16}$
	temps	0.636320	0.510569	1.740970
1.5	déterminant	$7.6639 * 10^{16}$	$1.5426 * 10^{17}$	$7.6639 * 10^{16}$
	temps	0.531401	0.501505	1.908095

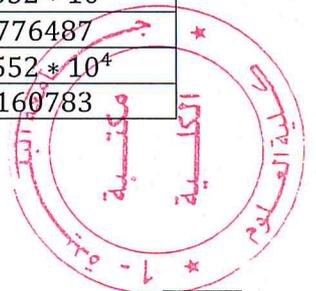
Tableau 17 : plan avec 3 facteurs, modèle de 2^{ème} degré

Cas d'un plan avec 3 facteurs : $k = 3$ $N = 69$ $n = 12$ $p = 10$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$4.9552 * 10^4$	$4.9552 * 10^4$	$4.9552 * 10^4$
	temps	0.682217	0.600580	1.832047
0.5	déterminant	$4.9552 * 10^4$	$4.9552 * 10^4$	$4.9552 * 10^4$
	temps	0.763074	0.736436	1.589213
1	déterminant	$4.9552 * 10^4$	$4.9552 * 10^4$	$4.9552 * 10^4$
	temps	0.801920	0.771957	1.776487
1.5	déterminant	$4.9552 * 10^4$	$4.9552 * 10^4$	$4.9552 * 10^4$
	temps	0.841403	0.658307	1.160783

Tableau 18 : plan avec 3 facteurs, modèle de 3^{ème} degré



Cas d'un plan avec 4 facteurs : $k = 4$ $N = 26$ $n = 12$ $p = 5$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$9.2586 * 10^3$	$9.2586 * 10^3$	$9.2586 * 10^3$
	temps	0.928324	0.855103	2.401122
0.5	déterminant	$9.2586 * 10^3$	$9.2586 * 10^3$	$9.2586 * 10^3$
	temps	0.912884	0.851436	1.981647
1	déterminant	$9.2586 * 10^3$	$9.2586 * 10^3$	$9.2586 * 10^3$
	temps	0.963064	0.661615	1.221047
1.5	déterminant	$9.2586 * 10^3$	$9.2586 * 10^3$	$9.2586 * 10^3$
	temps	0.794623	0.812586	1.048974

Tableau 18 : plan avec 4 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 4 facteurs : $k = 4$ $N = 32$ $n = 15$ $p = 15$

Modèle de 2^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$6.1052 * 10^{25}$	$6.1052 * 10^{25}$	$6.1052 * 10^{25}$
	temps	1.754153	1.930273	3.853034
0.5	déterminant	$6.1052 * 10^{25}$	$1.2555 * 10^{26}$	$6.1052 * 10^{25}$
	temps	2.070369	0.121274	3.041055
1	déterminant	$3.5705 * 10^{26}$	$1.2555 * 10^{26}$	$6.1052 * 10^{25}$
	temps	2.057226	0.132932	2.469058
1.5	déterminant	$1.7511 * 10^{27}$	$1.2328 * 10^{26}$	$6.1052 * 10^{25}$
	temps	0.149823	0.129551	1.855423

Tableau 19 : plan avec 4 facteurs, modèle de 2^{ème} degré

Cas d'un plan avec 4 facteur : $k = 4$ $N = 26$ $n = 12$ $p = 8$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$7.4139 * 10^3$	$7.4139 * 10^3$	$7.4139 * 10^3$
	temps	0.877648	0.982504	2.270073
0.5	déterminant	$7.4139 * 10^3$	$1.5572 * 10^4$	$7.4139 * 10^3$
	temps	1.065952	0.530751	1.713267
1	déterminant	$7.4139 * 10^3$	$1.5572 * 10^4$	$7.4139 * 10^3$
	temps	0.979931	0.523830	1.523353
1.5	déterminant	$7.4139 * 10^3$	$7.4139 * 10^3$	$7.4139 * 10^3$
	temps	1.082275	1.097080	0.976198

Tableau 20 : plan avec 4 facteurs, modèle de 3^{ème} degré

Cas d'un plan avec 5 facteurs : $k = 5$ $N = 30$ $n = 12$ $p = 6$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$5.5602 * 10^9$	$5.5602 * 10^9$	$5.5602 * 10^9$
	Temps	1.243231	1.024202	1.494009
0.5	déterminant	$5.5602 * 10^9$	$5.5602 * 10^9$	$5.5602 * 10^9$
	Temps	1.135074	1.018004	1.199934
1	déterminant	$5.5602 * 10^9$	$5.5602 * 10^9$	$5.5602 * 10^9$
	Temps	1.168445	1.040812	1.198872
1.5	déterminant	$5.5602 * 10^9$	$5.5602 * 10^9$	$5.5602 * 10^9$
	Temps	1.197990	1.080666	1.148107

Tableau 21 : plan avec 5 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 5 facteurs : $k = 5$ $N = 30$ $n = 10$ $p = 10$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$8.2720 * 10^{13}$	$8.2720 * 10^{13}$	$8.2720 * 10^{13}$
	temps	1.346295	1.186397	2.164667
0.5	déterminant	$2.8817 * 10^{14}$	$5.5915 * 10^{14}$	$8.9720 * 10^{13}$
	temps	1.399747	0.003269	1.437538
1	déterminant	$4.4603 * 10^{14}$	$5.2266 * 10^{14}$	$2.3188 * 10^{14}$
	temps	1.401015	0.003064	1.841945
1.5	déterminant	$4.7979 * 10^{14}$	$5.5915 * 10^{14}$	$3.1693 * 10^{14}$
	temps	1.373336	0.004074	1.490520

Tableau 22 : plan avec 5 facteurs, modèle de 3^{ème} degré

Cas d'un plan avec 6 facteurs : $k = 6$ $N = 21$ $n = 8$ $p = 7$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$1.1352 * 10^9$	$1.1352 * 10^9$	$1.1352 * 10^9$
	Temps	0.963464	0.862839	1.025431
0.5	déterminant	$1.1352 * 10^9$	$3.1238 * 10^9$	$1.1352 * 10^9$
	Temps	0.949433	0.144804	1.042654
1	déterminant	$1.1352 * 10^9$	$3.0487 * 10^9$	$1.1352 * 10^9$
	Temps	0.924856	0.118777	0.961862
1.5	déterminant	$1.1352 * 10^9$	$4.0038 * 10^9$	$1.1352 * 10^9$
	Temps	0.987938	0.019650	1.086261

Tableau 23 : plan avec 6 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 6 facteurs : $k = 6$ $N = 21$ $n = 12$ $p = 12$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$9.4142 \cdot 10^{17}$	$9.4142 \cdot 10^{17}$	$9.4142 \cdot 10^{17}$
	temps	1.864336	1.882042	2.365684
0.5	déterminant	$2.1799 \cdot 10^{18}$	$3.5999 \cdot 10^{18}$	$9.4142 \cdot 10^{17}$
	temps	1.923150	0.005530	2.047421
1	déterminant	$1.5750 \cdot 10^{19}$	$3.5999 \cdot 10^{18}$	$9.4142 \cdot 10^{17}$
	temps	1.760359	0.005317	1.930131
1.5	déterminant	$6.6939 \cdot 10^{19}$	$2.1922 \cdot 10^{18}$	$9.4142 \cdot 10^{17}$
	temps	1.941584	0.002637	1.678409

Tableau 24 : plan avec 6 facteurs, modèle de 3^{ème} degré

Cas d'un plan avec 7 facteurs : $k = 7$ $N = 21$ $n = 10$ $p = 8$

Modèle de 1^{er} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$4.0950 \cdot 10^{10}$	$4.0950 \cdot 10^{10}$	$4.0950 \cdot 10^{10}$
	Temps	1.336062	1.094079	1.455698
0.5	déterminant	$4.0950 \cdot 10^{10}$	$2.6226 \cdot 10^{11}$	$4.0950 \cdot 10^{10}$
	Temps	1.125368	0.28455	1.415122
1	déterminant	$4.0950 \cdot 10^{10}$	$2.6512 \cdot 10^{11}$	$4.0950 \cdot 10^{10}$
	Temps	1.321988	0.019113	1.390632
1.5	déterminant	$4.0950 \cdot 10^{10}$	$2.2803 \cdot 10^{11}$	$4.0950 \cdot 10^{10}$
	Temps	1.313361	0.008625	1.332306

Tableau 25 : plan avec 7 facteurs, modèle de 1^{er} degré

Cas d'un plan avec 7 facteurs : $k = 7$ $N = 31$ $n = 14$ $p = 14$

Modèle de 3^{ème} degré

alpha		Donev	Donev modifié	Fdop modifié
0.1	déterminant	$1.8155 \cdot 10^{26}$	$1.8155 \cdot 10^{26}$	$1.8155 \cdot 10^{26}$
	temps	2.490877	2.054707	3.446412
0.5	déterminant	$1.8155 \cdot 10^{26}$	$4.2609 \cdot 10^{26}$	$1.8155 \cdot 10^{26}$
	temps	2.267786	0.968066	2.385057
1	déterminant	$1.8155 \cdot 10^{26}$	$5.9665 \cdot 10^{26}$	$1.8155 \cdot 10^{26}$
	temps	2.515117	0.397260	2.744391
1.5	déterminant	$1.8155 \cdot 10^{26}$	$4.1023 \cdot 10^{26}$	$1.8155 \cdot 10^{26}$
	temps	2.550490	0.208976	2.283648

Tableau 26 : plan avec 7 facteurs, modèle de 3^{ème} degré

CONCLUSION

Au cours de notre travail, nous avons utilisé les algorithmes d'échange (Fedorov, Fedorov modifié, k-échange, kl-échange, kl-échange modifié, Fdop) et de déplacement (Fdop modifié, Donev).

Ces algorithmes ont été programmé à l'aide de MATLAB permettent une comparaison entre eux (valeur de déterminant et au temps d'exécution). Le meilleur algorithme s'avère être celui de Fedorov modifié (pour les algorithmes d'échange) et celui de Fdop modifié (pour les algorithmes de déplacement).

Enfin, nous avons effectué une modification de l'algorithme de DONEV en obtenant de meilleurs résultats que précédemment.

Les matrices

❖ Cas d'un plan avec 2 facteurs

Modèle de 1^{ère} degré

1	1
-1	1
0	-1

Modèle de 2^{ème} degré

-0.5	-1
-5	2
-5	6.8
3.7	6.8
3.7	-1
-5	-1
3.7	2
-5	6.8
-0.5	6.8
-0.5	4
3.7	6.8

Modèle de 3^{ème} degré

-0.5	-1
-5	2
-5	6.8
3.7	6.8
3.7	-1
-5	-1
3.7	2
-5	6.8
-0.5	6.8
-0.5	4
3.7	6.8

❖ Cas d'un plan avec 3 facteurs

Modèle de 1^{ère} degré

1	0.5	1
1	-1	1
-1	1	1
-1	-1	-1
1	-1	1
1	-1	-1
-1	-1	1
1	0.5	-1
-1	-1	-1
-1	1	-1
1	0.5	-1
-1	1	1

Modèle de 2^{ème} degré

1	0.5	-1
1	-1	1
-1	-1	-1
1	-1	-1
-1	1	-1
-1	-1	1
-1	1	1
1	0.5	1
-1	0	0
0	0	1
0	0	-1
0	1	0

Modèle de 3^{ème} degré

1	0.5	1
1	-1	-1
0	-1	1
-1	0	1
0	1	1
-1	1	1
0	0	-1
1	0	-1
-1	1	-1
1	-1	1
0	1	-1
-1	-1	-1

❖ Cas d'un plan avec 4 facteurs

Modèle de 1^{ère} degré

-1	0	0	1
1	0	0	1
0	-0.577	0.816	1
0.5	0.866	0	1
0.5	-0.866	0	-1
-0.5	-0.866	0	1
-0.5	0.289	0.819	-1
0	0.577	0.866	-1
0.5	-0.866	0	-1
-0.5	-0.289	-0.816	-1
0	0.577	-0.876	-1
0	0.577	-0.816	1

Modèle de 2^{ème} degré

2	-1.5	-3	6
2	0.5	4	6
1.7	1.5	4	4.8
-2	-1.5	-1.4	0.9
2	1	1.4	4.2
-1	1.5	4	6
-2	1.5	-1.4	1.2
-2	1	4	0.9
2	1.5	-3	5.5
1	-1.5	-3	1.2
1	-1.5	4	0
1	1	-3	0

1	1	-1.4	6
2	-1.5	4	5.5
-1	1.5	-3	6

Modèle de 3^{ème} degré

1	0	0	1
0	-0.577	0.816	1
-0.5	0.866	0	1
-1	0	0	-1
0.5	-0.866	0	-1
0	0	0	-1
-1	-0.866	-0.875	1
0	0.577	0.866	-1
0.5	0.866	0	-1
0.5	-0.289	-0.816	1
0	0.577	-0.876	-1
0	0	0	1

❖ Cas d'un plan avec 5 facteurs

Modèle de 1^{ère} degré

2	1.5	-3	6	-2
-2	-1.5	-3	1	-5
2	-1.5	4	6	4
1	1.8	-3	2	-3
2	-1.5	-3	6	4
-2	1.8	4	3	-5
-2	-1.5	-3	1	-5
-2	1.5	-3	3	3
-2	1.5	4	3	3

1	1.8	4	6	-3
1	-1.5	1.4	0	-3
1	-1.5	1.4	0	-3

Modèle de 3^{ème} degré

2	0	-3	6	-5
1	1.8	-1.4	6	-3
2	-1.5	-1.4	6	4
-1	1.5	-3	6	4
-2	-1.5	-3	1	-5
2	1.8	4	6	2.5
-2	1.5	-1.4	3	3
1	1.8	-3	2	-3
1	-1.5	1.4	0	-3
1	-1.5	4	2	-3

❖ Cas d'un plan avec 6 facteurs

Modèle de 1^{ère} degré

2	1.8	4	0	4	2.5
1	1.8	-1.4	6	-3	-0.5
-1	1.5	-3	6	4	0.5
-2	1.5	4	3	3	-2.5
1	-1.5	-3	2	3	-2.5
2	-1.5	4	6	4	0.5
1	-1.5	1.4	0	-3	-0.5
-2	1.8	4	3	-5	-2.5

Modèle de 3^{ème} degré

-1.77	1.8	4	1.5	2.5	2
0.99	-1.5	1.5	2.7	0.54	-5
0.23	0	1.4	0	2.5	-5
11	-1.5	1.4	0	-1	0
10.1	1.8	4	-1.22	-3	1
10.1	1.8	-3	2	-0.5	0
0	-1.5	1.6	2.5	-1	0
-1.9	1.5	-3	-1.22	0.85	1
-1.9	-1	4	-1.22	0.85	1
1.7	-1.5	-3	2.7	0.85	1
1.7	-1.5	1.6	2.7	0.54	2
0.23	0	-3	2.7	2.5	-2.5

❖ Cas d'un plan avec 7 facteurs

Modèle de 1^{ère} degré

-1	-1.5	-2	6	-5	-2.5	7
-1	1.5	-3	6	4	0.5	2.5
-2	1.8	4	3	-5	-2.5	1
-2	1.5	4	3	3	-2.5	1
1	-1.5	-3	2	3	-2.5	1
2	1.8	4	6	2.5	2.5	7
1	-1.5	1.4	0	-3	-0.5	1
1	1.8	-3	2	-3	-0.5	2.5
2	-1.5	-3	6	4	0.5	2.5
1	1.8	4	6	-3	0.5	2.5

Modèle de 3^{ème} degré

0.99	-1.5	1.5	2.7	0.54	-5	-0.7
1	-1.5	-3	2	3	-2.5	3.1
-2	-1.45	-0.25	-1	-5	-5	0.15
1	1.8	-1.4	6	-3	-0.5	0
11	-1.5	1.4	0	-1	0	2.33
-1.9	1.5000	-3	-1.22	0.85	1	1.15
10.1	1.6	4	-1.22	-3	1	1.15
-1.9	-1	4	-1.22	0.85	1	-2
-2	1.5	4	3	3	-1.5	1.4
1.24	1.5	-1.4	0.4	2.5	-2.5	-1.36
0.99	-1.5	-1.4	2.7	0.54	3.2	-0.7
1	1.8	-3	2	-3	-0.5	2.66
0.23	0	1.4	0	2.5	-2.5	3.22
0.99	0	-3	2.9	2.5	-5	

Les programmes sur matlab

Ces algorithmes sont codés et exécutés sur un PC équipé d'un processeur pentium 3.133 MHz avec mémoire de 4 GB de RAM sous Windows 7. Les algorithmes implémentés en MATLAB.

➤ FEDEROV :

```
model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré tapez 2');
N=input('donnez N');
n=input('donnez n');
y=input('donnez la matrice des pt candidats');
switch model
    case 1
        Y=premddeg(y);
    case 2
        Y=deuxdegcomp(y);
    otherwise
        disp('erreur');
end
[l,coef]=size(Y);
X=ptdapui(Y,n,N,coef);
p=1;
it=0;
tic
while ((p>0)&&(it<2000))
    invers=(inv(X'*X));
max=-1000;
for i=1:n
    dxi=X(i,:)*invers*(X(i,:))';
    for j=1:N
        dxj=Y(j,:)*invers*(Y(j,:))';
        dxij=Y(j,:)*invers*(X(i,:))';
        D=dxj-(dxi*dxj-dxij^2)-dxi;
        if (D>max)
            max=D;
            s=i;
            e=j;
        end
    end
end
p=-1;
if(max>10^-6)
    X(s,:)=Y(e,:);
    p=1;
    it=it+1;
end
end
disp('it');
toc
dt=det(X'*X);
disp(dt);
```

➤ **FEDEROV modifier :**

```
model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré tapez 2 deuxième degré réduit tapez 3');
N=input('donnez N');
n=input('donnez n');
y=input('donnez la matrice des pt candidats');
ech=0;
switch model
    case 1
        Y=premddeg(y);
    case 2
        Y=deuxdegcomp(y);
    case 3
        Y=deuxdeg(y);
    otherwise
        disp('erreur');
end
[l,coef]=size(Y);
X=ptdapui(Y,n,N,coef);
p=1;
it=0;
tic
while ((p==1)&&(it<200))
max=-1000;
p=-1;
for i=1:n
    invers=inv(X'*X);
    dxi=X(i,:)*invers*(X(i,:))';
    for j=1:N
        dxj=Y(j,:)*invers*(Y(j,:))';
        dxij=Y(j,:)*invers*(X(i,:))';
        D=dxj-(dxi*dxj-dxij^2)-dxi;
        if (D>max)
            max=D;
            s=i;
            e=j;
        end
    end
end
if(max>10^-6)
    for c=1:coef
        X(s,c)=Y(e,c);
    end
    ech=ech+1;
    p=1;
    it=it+1;
end
max=-1000;
end
end
toc
dt=det(X'*X);
disp(dt);
```

➤ K-échange :

```
model=input('si vous avez utiliser un modèle du premier degré tapez 1
deuxième degré tapez 2');
N=input('donnez N');
n=input('donnez n');
k=input('donnez k');
y=input('donnez la matrice des pt candidats');
switch model
    case 1
        Y=premdeg(y);
    case 2
        Y=deuxdegcomp(y);
    otherwise
        disp('erreur');
end
[l,coef]=size(Y);
Xk=kpoint(Y,k,n,N);
p=1;
it=0;
tic
while ((p==1)&&(it<200))
max=-1000;
p=-1;
for i=1:k
    invers=inv(Xk'*Xk);
    dxi=Xk(i,:)*invers*(Xk(i,:))';
    for j=1:N
        dxj=Y(j,:)*invers*(Y(j,:))';
        dxij=Y(j,:)*invers*(Xk(i,:))';
        D=dxj-(dxi*dxj-dxij^2)-dxi;
        if (D>max)
            max=D;
            s=i;
            e=j;
        end
    end
end
if(max>10^-6)
    for c=1:coef
        Xk(s,c)=Y(e,c);
    end
    p=1;
    it=it+1;
end
max=-1000;
end
end
toc
dt=det(Xk'*Xk);
disp(dt);
```

➤ KL-échange :

```
model=input('si vous avez utiliser un modèle du premier degré tapez 1
deuxième degré tapez 2');
N=input('donnez N');
n=input('donnez n');
k=input('donnez k');
l=input('donnez l');
y=input('donnez la matrice des pt candidats');
switch model
    case 1
        Y=premdeg(y);
    case 2
        Y=deuxdegcomp(y);
    otherwise
        disp('erreur');
end
[l,coef]=size(Y);
Xk=kpoint(Y,k,n,N);
yl=lpoint(Y,Xk,l,N);
disp(Xk);
disp(yl);
p=1;
it=0;
tic
while ((p==1)&&(it<200))
    max=-1000;
    p=-1;
    for i=1:k
        invers=inv(Xk'*Xk);
        dxi=Xk(i,:)*invers*(Xk(i,:))';
        for j=1:l
            dxj=yl(j,:)*invers*(yl(j,:))';
            dxij=yl(j,:)*invers*(Xk(i,:))';
            D=dxj-(dxi*dxj-dxij^2)-dxi;
            if (D>max)
                max=D;
                s=i;
                e=j;
            end
        end
    end
    if(max>10^-6)
        for c=1:coef
            Xk(s,c)=Y(e,c);
        end
        p=1;
        it=it+1;
    end
end
max=-1000;
end
end
toc
dt=det(Xk'*Xk);
disp(dt);
```

➤ FDOP :

```
model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré tapez 2');
N=input('donnez N');
n=input('donnez n');
R=input('donnez R');
M=input('donnez M');
y=input('donnez la matrice des pt candidats');
ptm=input('donnez la position de M point qui construire R dans une
matrice');
switch model
case 1
    Y=premddeg(y);
case 2
    Y=deuxdegcomp(y);
otherwise
    disp('erreur');
end
[l,coef]=size(Y);
X=ptdapui(Y,n,N,coef);
t=0;
it=0;
ech=0;
tic
while((t<4)&&(it<100))
    invers=inv(X'*X);
    x=X;
    for j=1:R
        xm=Y(ptm(j,:),:);
        Xj=[X;xm];
        i=M;
        while(i~=0)
            min=Xj(1,:)*invers*(Xj(1,:))';
            s=1;
            for k=2:n+i
                dx=Xj(k,:)*invers*(Xj(k,:))';
                if(dx<min)
                    min=dx;
                    s=k;
                end
            end
            Xj=[Xj(1:s-1,:);Xj(s+1:n+i,:)];
            i=i-1;
        end
        if(det(X'*X)<det(Xj'*Xj))
            X=Xj;
        end
    end
    if(det(x'*x)<det(X'*X))
        t=t+1;    ech=ech+1;
    else
        t=0;
    end
end
it=it+1;
end
d=det(X'*X); disp(d);
toc
```

➤ **FDOP modifier :**

```
model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré complet tapez 2 deuxième degré tapez 3');
n=input('donnez n');
R=input('donnez R');
M=input('donnez M');
x=input('donnez X');
ptm=input('donnez la position de M point qui construire R dans une
matrice');
switch model
    case 1
        X=premdeg(x);
    case 2
        X=deuxdegcomp(x);
    case 3
        X=deuxdeg(x);
    otherwise
        disp('erreur');
end
alpha=input('donnez alpha');
[l,cln]=size(X);
t=0;
it=0;
ech=0;
tic
while((t<4)&&(it<100))
    x=X;
    for j=1:R
        Xj=X;
        invers=inv(Xj'*Xj);
        for i=1:M
            D=zeros(1,cln);
            D(1,2)=1;
            Xp=x(ptm(j,i),:);
            Xp=Xp+alpha*D;
            Xj(n+i,:)=Xp(1,:);
        end
        while(i~=0)
            min=Xj(1,:)*invers*(Xj(1,:))';
            s=1;
            for k=2:n+i
                dx=Xj(k,:)*invers*(Xj(k,:))';
                if(dx<min)
                    min=dx;
                    s=k;
                end
            end
            Xj=[Xj(1:s-1,:);Xj(s+1:n+i,:)];
            i=i-1;
        end
        if(det(X'*X)<det(Xj'*Xj))
            X=Xj;
            ech=ech+1;
        end
    end
end
if(det(x'*x)<det(X'*X))
    t=t+1;
else
    t=0;
end
end
```

```

it=it+1;
end
toc
d=det(X'*X);

```

➤ **DONEV :**

```

model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré complet tapez 2 deuxième degré réduit tapez 3');
n=input('donnez n');
alpha=input('donnez alpha');
alphamin=input('donnez alpha min');
x=input('donnez la matrice des pt candidats');
switch model
case 1
X=premdeg(x);
case 2
X=deuxdegcomp(x);
case 3
X=deuxdeg(x);
otherwise
disp('erreur');
end
[l,cln1]=size(x);
[l,cln2]=size(X);
it=0;
ech=0;
tic
while((alpha>alphamin)&&(it<1000))
deltamax=0;
invers=(inv(X'*X));
for i=1:n;
di=X(i,:)*invers*(X(i,:))';
for j=2:cln1+1;
for sens=[-1 1];
D=zeros(1,cln2);
D(1,j)=sens;
Xp=X(i,:);
Xp=Xp+alpha*D;
dp=Xp*invers*(Xp)';
dip=Xp*invers*(X(i,:))';
delta=dp-(di*dp-dip^2)-di;
if(delta>deltamax)
deltamax=delta;
s=i;
xprim=Xp;
end
end
end
if(deltamax>1)
X(s,:)=xprim;
ech=ech+1;
else
alpha=alpha/2;
end
it=it+1;
end
toc
dt=det(X'*X);

```

➤ **DONEV modifier :**

```
model=input('si vous avez utiliser modèle de premier degré tapez 1
deuxième degré complet tapez 2 deuxième degré tapez 3 ');
n=input('donnez n');
a=input('donnez alpha');
alphamin=input('donnez alpha min');
x=input('donnez la matrice des pt candidats');
switch model
    case 1
        X=premdeg(x);
    case 2
        X=deuxdegcomp(x);
    case 3
        X=deuxdeg(x);
    otherwise
        disp('erreur');
end
[l,cln1]=size(x);
[l,cln2]=size(X);
it=0;
ech=0;
tic
for i=1:5
    deltamax=0;
    alpha=a;
    invers=(inv(X'*X));
    di=X(i,:)*invers*(X(i,:))';
    xt=X(i,:);
    while((alpha>alphamin)&&(it<1000))
        for j=2:cln1+1;
            for sens=[-1 1];
                D=zeros(1,cln2);
                D(1,j)=sens;
                Xp=xt;
                Xp=Xp+alpha*D;
                dp=Xp*invers*(Xp)';
                dip=Xp*invers*(X(i,:))';
                delta=dp-(di*dp-dip^2)-di;
                if(delta>deltamax)
                    deltamax=delta;
                    xprim=Xp;
                end
            end
        end
        if(deltamax<1)
            alpha=alpha/2;
        end
        it=it+1;
        xt=xprim;
    end
    if(deltamax>1)
        X(i,:)=xprim;
        ech=ech+1;
    end
end
end
toc
dt=det(X'*X);
disp(dt);
```

les fonctions utiliser dans la programmation des algorithmes

➤ ptdapui :

```
function X = ptdapui(Y,n,N,coef)
ptl=1;
ptn=n;
d=0;
while((d<=10^-16)&&(ptn<=N))
    if(ptn<N)
        for i=ptl:ptn
            for j=1:coef
                X(i-ptl+1,j)=Y(i,j);
            end
        end
        d=det(X'*X);
        if(d<=10^-6)
            ptl=ptl+1;
            ptn=ptn+1;
        end
    else
        ptdap=input('donnez les point qui construire X dans un
vecteur');
        for i=ptdap
            X=Y(ptdap,:);
        end
        d=det(X'*X);
        disp(d);
    end
end
```

➤ premdeg :

```
function Y = premdeg(y)
Y=[ones(size(y(:,1))),y];
```

➤ deuxdegcomp :

```
function Y = deuxdegcom(y)
[lgn,cln]=size(y);
rctngl=[];
car=[];
for i=1:cln
    for j=i+1:cln
        rctngl=[rctngl,y(:,i).*y(:,j)];
    end
end
for i=1:cln
    car=[car,y(:,i).*y(:,i)];
end
Y=[ones(size(y(:,1))),y,rctngl,car];
```

➤ **deuxdeg :**

```
function Y = deuxdeg(y)
[lgn,cln]=size(y);
car=[];
for i=1:cln-1
    car=[car,y(:,i).*y(:,i)];
end
Y=[ones(size(y(:,1))),y,car];
```

➤ **Kpoint :**

```
function Xk = kpoint(Y,k,n,N)
d=0;
ptn=n;
ptl=1;
while(d<=10^-6)
    if(ptn<N)
        for i=ptl:ptn
            X(i-ptl+1,:)=Y(i,:);
        end
    else
        ptdap=input('donnez les point qui construire X dans un
vecteur');
        for i=ptdap
            X=Y(ptdap,:);
        end
    end
    for i=1:n
        x(i,:)=X(i,:);
    end
    invers=inv(X'*X);
    for j=1:k
        dxj=x(j,:)*invers*(x(j,:))';
        min=dxj;
        c=0;
        for i=j+1:n
            dxi=x(i,:)*invers*(x(i,:))';
            if(dxi<min)
                min=dxi;
                ligne=i;
                c=1;
            end
        end
        if(c==1)
            sauv=x(j,:);
            x(j,:)=x(ligne,:);
            x(ligne,:)=sauv;
        end
    end
    for i=1:n
        Xk(i,:)=x(i,:);
    end
    disp(Xk);
    d=det(Xk'*Xk);
    disp(d);
    ptl=ptl+1;
    ptn=ptn+1;
end
```

➤ **lpoint :**

```
function yl = lpoint(Y,X,l,N)
for i=1:N
    y(i,:)=Y(i,:);
end
invers=inv(X'*X);
for j=1:l
    dyj=y(j,:)*invers*(y(j,:))';
    max=dyj;
    c=0;
    for i=j+1:N
        dyi=y(i,:)*invers*(y(i,:))';
        if(dyi>max)
            max=dyi;
            ligne=i;
            c=1;
        end
    end
end
if(c==1)
    sauv=y(j,:);
    y(j,:)=y(ligne,:);
    y(ligne,:)=sauv;
end
end
for i=1:N
    yl(i,:)=y(i,:);
end
end
```

REFERENCES

[1] Stéphane Vivier. *Stratégies d'optimisation par la méthode des plans d'expériences et Application aux dispositifs électrotechniques modélisés par éléments finis*. Doctorat délivré conjointement par l'Ecole centrale de Lille et l'Université des sciences et technologies de Lille. 2002.

[2] Pierre Franquart. *Optimisations multi-critères et Méthodologie de la recherche expérimentale*. Thèse pour obtenir le grade de Docteur en sciences de l'université de droit, d'économie et des sciences d'AIX-MARSEILLE 3. 1992.

[3] Anne Peissik. *Méthodologie de la recherche expérimentale propriétés et caractéristique des matrices d'expériences pour les modèles polynomiaux du second degré*. Thèse pour obtenir le grade de Docteur en sciences de l'université de droit, d'économie et des sciences d'AIX-MARSEILLE 3. 1995.

[4] Maurice Pillet. *Les plans d'expériences par la méthode TAGUCHI*. Les éditions D'organisation.

[5] Fabian Triefenbach, *Design of Experiments: The D-Optimal Approach and Its Implementation As a Computer Algorithm*, 2008.

[6] Stéphane GAZUT, *Conception et mise en œuvre de nouvelles méthodes d'élaboration de plans d'expériences pour l'apprentissage de modèles non linéaires*, Thèse de doctorat, Université PARIS-SUD XI, Faculté des sciences d'Orsay, 2007.

[7] Philippe TRIBOULET (Lycée Niepce – Chalon sur Saône) 09/09/2008.

[8] Nam-Ky Nguyen and Greg F. Piepel, *Computer-Generated Experimental Designs for Irregular-Shaped Regions*, School of Mathematics, Statistics and Computer Science, University of New England, Armidale, NSW, Australia² Statistical Sciences, Battelle-Pacific Northwest.

[9] Jacques Goupy. *Plans d'expériences pour surfaces de réponse : Technique et ingénierie*, Série Génie industriel. Dunod.2009.

