UNIVERSITÉ DE MONTRÉAL

IDENTIFICATION STATISTIQUE DE VARIABLES IMPORTANTES POUR L'OPTIMISATION DE BOÎTES NOIRES

IMEN BEN YAHIA
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AOÛT 2012

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

IDENTIFICATION STATISTIQUE DE VARIABLES IMPORTANTES POUR L'OPTIMISATION DE BOÎTES NOIRES

présenté par: BEN YAHIA Imen
en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. PARTOVI NIA Vahid, Ph.D., président
M. AUDET Charles, Ph.D., membre et directeur de recherche
M. ADJENGUE Luc, Ph.D., membre et codirecteur de recherche
M. ANJOS Miguel, Ph.D., membre

REMERCIEMENTS

Je tiens à remercier sincèrement tous ceux qui m'ont aidé à réaliser ce travail à commencer par mes directeurs de recherche Charles Audet et Luc Adjengue pour leur support et leur disponibilité.

J'exprime ma gratitude à Charles Audet, Sébastien Le Digabel, Christophe Tribes ainsi qu'au personnel du GERAD pour l'aide technique qu'ils m'ont apportée.

J'adresse également mes remerciements aux membres du jury qui ont accepté d'évaluer ce travail.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

RÉSUMÉ

De nombreux problèmes industiels possèdent des propriétés particulières nécessitant le recours aux méthodes d'optimisation sans dérivées qui ont connu un accroissement d'intérêt au cours des dernières années. C'est notamment le cas lorsque les fonctions et les contraintes définissant le problème d'optimisation sont non linéaires, non différentiables, bruitées ou non définies pour certains points du domaine. Les méthodes de recherche directes tel que GPS et MADS sont des méthodes sans dérivées qui s'intéressent à résoudre, sous contraintes, des problèmes d'optimisation de boîtes noires à simple objectif ou biobjectif, où les fonctions correspondent le plus souvent au résultat d'un code informatique. Les sorties des boîtes noires sont parfois très coûteuses à évaluer et peuvent échouer à retourner une valeur pour des raisons inconnues. C'est la principale motivation qui nous incite à restreindre le nombre d'appels à la boîte noire en utilisant une stratégie de fixation de variables qui nous ramènera à optimiser dans des sous-espaces. Nous proposons une méthode qu'on appellera STATS-MADS fondée sur l'analyse de sensibilité afin de classer les variables d'entrée selon leur impact sur la sortie. L'optimisation se fera en alternance entre l'espace des variables d'entrée et les sous-espaces obtenus en fixant les variables jugées moins influentes. La même technique est utilisée pour s'attaquer aux problèmes de grande dimension qui constituent une des limites des méthodes sans dérivées.

Nous utilisons la plus récente version ($3.5.2$) du logiciel NOMAD qui est une implémentation en C++ des algorithmes MADS, principalement l'instanciation ORTHOMADS. À la lumière des problèmes tests ayant jusqu'à 500 variables, nous comparons les résultats de notre méthode avec MADS et GPS afin de pouvoir conclure à son efficacité.

ABSTRACT

Many industial problems have particular features requiring the recourse to derivative-free optimization methods which have shown increasing interest in recent years. This is particularly the case when the functions and the constraints defining the optimization problem are nonlinear, nondifferentiable, noisy or not defined for some points of the domain. Direct search methods such as GPS and MADS are derivative-free methods interested in solving, under constraints, simple or biojectif blackbox problems where functions are usually the result of a computer code. The outputs of blackboxes may be very costly to evaluate and may fail to return a value for unknown reasons. This is the main motivation that drives us to reduce the number of calls to the blackbox by using a strategy of setting variables leading to optimize in subspaces. We propose a method which will be called STATS-MADS based on sensitivity analyses to rank the input variables according to their impact on the output. The optimization occurs by alternating between the whole space of input variables and the subspaces obtained by setting the less important variables. The same technique is used to tackle large-sized problems which is one of the limitations of derivative-free methods.

We use the most recent version (3.5.2) of the software NOMAD which is a C ++ implementation of MADS algorithms, mainly the instance ORTHOMADS. We compare the results of our method with MADS and GPS in order to conclude its effectiveness, based on test problems with up to 500 variables.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES NOTATIONS ET DES SYMBOLES

ANOVA	Analysis Of Variance.
AS	Analyse de Sensibilité.
CS	Coordinate Search.
DFO	Derivative-Free Optimization.
EFAV	Équation fondamentale de l'analyse de la variabilité.
FAST	Fast Fourier Transform.
GPS	Generalized Pattern Search.
HDMR	High Dimensional Model Representation.
LTMADS	Lower Triangular MADS.
MADS	Mesh Adaptive Direct Search.
MADS-RUN	Un appel à un MADS classique.
MADS-RESCUE	Un appel à MADS en cas de stagnation sous certaines circonstances .
NOMAD	Nonlinear Optimization with the MADS Algorithm.
OSD	Optimisation sans dérivées.
OAT	One At a Time.
ORTHOMADS	MADS avec directions orthogonales.
PSD-MADS	Parallel Space Decomposition with MADS.
STATS-MADS	MADS jumelé avec une méthode statistique.
SRC	Standardized Regression Coefficient.
SRRC	Standardized Rank Regression Coefficient.
EB	Extreme Barrier.
PB	Progressive Barrier.
PEB	Progressive to Extreme Barrier.

Ω	Domaine réalisable.
f	Fonction objectif.
f_Ω	Fonction objectif avec barrière sur Ω.
g	Fonction(s) des contraintes d'inégalité.
D	Ensemble générateur positif de $\mathbb{R}^n$.
$D\oplus$	Base positive maximale de $\mathbb{R}^n$.
Lb	Vecteur des bornes inférieures.
Ub	Vecteur des bornes supérieures.
k	Compteur d'itérations.
t	Point dans $\mathbb{R}^n$.
x_k	Centre de sonde à l'itération k.
D_k	Ensemble des directions de sonde à l'itération k.
M_k	Treillis conceptuel à l'itération k.
P_k	Ensemble à sonder à l'itération k.
S_k	Ensemble des points d'essai de l'étape de recherche à l'itération k.
Δ_k^m	Taille de treillis à l'itération k.
Δ_k^p	Taille de sonde à l'itération k.

LISTE DES ANNEXES

CHAPITRE 1

INTRODUCTION

L'optimisation est l'art consistant à maximiser ou minimiser des fonctions mathématiques souvent sujettes à des conditions particulières. Au 18ème siècle, le mathématicien suisse Leonhard Euler a proclamé que : ≪*...il n'arrive rien dans le monde qui ne présente quelque propriété de maximum ou de minimum.*≫ [1] De nos jours, l'optimisation envahit quasiment tous les domaines et son usage s'avère indispensable dans les sciences appliquées, l'ingénierie, la médecine, l'économie, etc.

Lorsque les fonctions régissant un système donné présentent des irrégularités, l'optimisation devient problématique. L'aspect non lisse en mathématiques et en optimisation est de plus en plus fréquent et dépeint un grand nombre de phénomènes naturels auxquels on fait face. Plusieurs problèmes de grande ampleur, dont nous en présenterons un, sont touchés par cet aspect aussi bien que par d'autres.

Ces problèmes s'écrivent sous la forme :

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} f(x)$$

où : $f : X \to \mathbb{R} \cup \{\infty\}$ est la fonction mathématique à minimiser ou *fonction objectif*, $\Omega = \{x \in X : g(x) \leq 0\}$ est le domaine réalisable, $g : X \to (\mathbb{R} \cup \{\infty\})^m$ sont les conditions à respecter ou *fonctions de contraintes*.

1. Les mathématiques, les idées et le réel physique (2006) par Lautman, p. 211 (Google-Livres)

1.1 Mise en contexte

Ce mémoire s'inscrit dans le cadre d'optimisation non lisse de boîtes noires. Ces dispositifs sont caractérisés par un fonctionnement interne dissimulé qui correspond, dans la plupart des cas, à un code informatique. Une boîte noire reçoit une ou plusieurs entrées et retourne une ou plusieurs sorties et peut être représentée shématiquement par la figure 1.1.

Figure 1.1 Schéma simplifié d'une boîte noire

Les sorties sont les fonctions objectif ou de contraintes qu'on désigne par fonctions de la boîte noire. On s'intéresse au cas *mono-objectif* (contraint ou non) pour lequel on cherche à minimiser une seule fonction. En effet, maximiser f revient à minimiser $-f$.
Mise à part leur caractère dissimulé, les boîtes noires peuvent se caractériser par :

- Des fonctions dont l'expression analytique n'est pas fournie et pouvant être discontinues, non différentiables et non convexes ;
- La présence possible de bruit qui ajoute de l'incertitude et de la rugosité ce qui perturbe les sorties ;
- Des échecs d'évaluation de fonctions sans motif (corruption dûe au bruit ou pour des raisons inconnues) dont le coût est souvent égal au coût de l'évaluation ;
- La multi-modalité : la présence de plusieurs optima locaux ;
- Un temps d'évaluation lourd qui peut varier de quelques minutes à quelques semaines ;
- Un espace mémoire important pour stocker les informations transmises (les sorties).

L'optimisation de boîtes noires est la tâche consistant à manipuler les entrées et lire les

sorties successivement jusqu'à l'obtention éventuelle d'une bonne solution et ce, sans aucune connaissance de l'intérieur de la boîte.

Un algorithme d'optimisation, par exemple MADS (Audet et Dennis, Jr., 2006) ou GPS (Torczon, 1997), est une automatisation de cette tâche. À partir d'un point initial $x_0 \in X$, on tente de trouver un nouveau candidat réduisant la fonction objectif f et respectant la contrainte $g(x) \leq 0$ (fig.1.2). Si c'est le cas, alors ce point devient le nouveau point à améliorer et les nouvelles valeurs de sortie sont retournées à l'algorithme d'optimisation, afin qu'il décide de retenir ou non un nouveau candidat. Les entrées envoyées à la boîte noire sont produites suivant une stratégie de recherche spécifique à chaque algorithme.

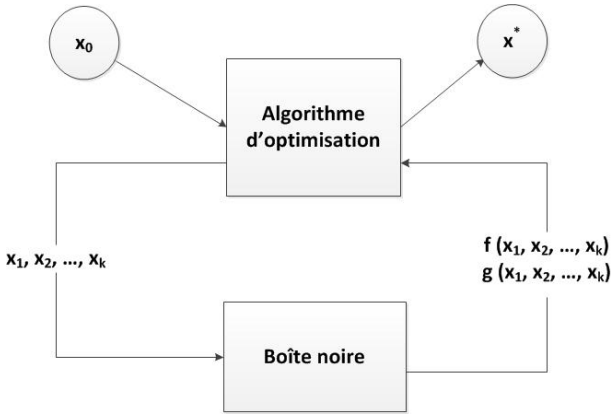

Figure 1.2 Principe d'un algorithme d'optimisation de boîtes noires pour un problème de dimension k à une seule contrainte

1.2 Motivations et spécifications

La modélisation de problèmes peut aboutir à des modèles de grande dimension pour lesquels les algorithmes d'optimisation deviennent impuissants, voire inopérants. En effet, l'explosion dimensionnelle est étroitement liée à l'inefficacité algorithmique : plus le volume de l'espace de recherche est grand, plus l'exploration de celui-ci devient difficile. Certains logiciels d'optimisation sont explicitement déconseillés au delà d'un certain

nombre de variables d'entrée. Il est donc crucial de trouver une manière pour manier cette situation.

Une stratégie de fixation de variables s'est révélée prometteuse en mode parallèle (Audet *et al.*, 2008c) et une autre en mode séquentiel (Booker *et al.*, 1998), en utilisant respectivement une méthode aléatoire et une méthode statistique afin de déterminer les variables à fixer. Cette approche a permis, dans les deux cas, une descente locale plus rapide (étant donné que l'exploration de l'espace de recherche est meilleure) et par conséquent une réduction du nombre d'appels à la boîte noire. Dans le présent document, on s'intéresse, d'une part, à la réduction du nombre d'évaluations des boîtes noires, indépendemment de la dimension de l'espace et d'autre part, à gérer le problème de la grande dimensionnalité. Nous nous inspirons de l'idée de la technique présentée dans (Booker *et al.*, 1998) qui repose sur une approche statistique qui a permis de filtrer 11 variables prépondérantes parmi 31 afin de développer une méthode faisant recours aux outils de la statistique pour l'identification de variables importantes. Nous nous sommes alors confrontés à deux difficultés majeures, à savoir : comment identifier les variables influentes (ce qui revient à identifier celles qui ne le sont pas) et quelle proportion fixer ?

La principale particularité de la technique que nous évoquons par rapport à celle mentionnée est qu'on fait appel à la méthode statistique d'une manière itérative, ce qui permet d'appliquer l'algorithme d'optimisation sur des sous-espaces de variables d'entrée différents.

Le principe de la technique est le suivant : on part d'un ensemble de variables d'entrée (espace complet), on lance l'algorithme d'optimisation sur cet ensemble. Par la suite, on cherche un sous-ensemble de variables importantes à l'aide de la méthode statistique sur lequel on relance l'algorithme et ainsi de suite.

Les données à partir desquelles on effectuera une analyse statistique sont regroupées dans un fichier historique qui recense toutes les solutions visitées par l'algorithme d'optimisation ainsi que les sorties de la boîte noire correspondantes. Dans le cadre de ce travail nous avons choisi d'implémenter la méthode statistique en C++ sur le logiciel

NOMAD (Le Digabel, 2011), pour laquelle l'identification de variables prépondérantes se fait en un temps de calcul raisonnable tout en tenant compte des caractéristiques des fonctions de la boîte noire.

1.3 Plan du mémoire

Les propos de ce mémoire sont étalés sur sept chapitres organisés de la manière suivante. Les deux premiers chapitres comportent deux revues de la littérature. L'une porte sur les méthodes d'analyse de sensibilité. Nous tenterons à travers celle-ci de trouver une méthode statistique adéquate permettant de classer les variables d'entrée, afin d'identifier les variables importantes. L'autre revue concerne les méthodes directes pour l'optimisation non lisse. On y retrouve le cadre théorique des algorithmes de recherche directe qui nous intéressent. Ainsi, nous serons en mesure d'intégrer l'aspect statistique à l'optimisation. Par la suite, nous présenterons une méthode statistique, conforme à nos spécifications, dont nous décrirons les différents ingrédients. Le chapitre cinq présente la structure algorithmique de notre méthode ainsi que les extensions qui en dérivent. Le chapitre six est consacré aux tests des boîtes noires et aux résultats numériques des différentes stratégies qui sont présentées plus tôt. Des profils de performance nous seront utiles pour apporter un jugement sur l'efficacité de chacune des stratégies. Le mémoire se conclut au septième chapitre dans lequel nous récapitulerons toutes les démarches et résultats importants.

CHAPITRE 2

REVUE DE LA LITTÉRATURE : MÉTHODES DIRECTES ET OPTIMISATION NON LISSE

Nous présentons d'abord le cadre général de notre projet en abordant le thème de l'optimisation non lisse et les méthodes qui y sont dédiées. Après avoir situé les méthodes de recherche directe dans leur contexte historique, nous en exposons quelques notions fondamentales. Finalement, nous détaillons les algorithmes de recherche directe qui nous intéressent pour la suite de ce mémoire, principalement pour des fins de comparaison.

2.1 Optimisation sans dérivées

L'optimisation sans dérivées (OSD) regroupe l'ensemble des méthodes typiquement désignées pour résoudre des problèmes d'optimisation, où l'estimation des dérivées (par exemple par différences finies (Dennis et Schnabel, 1996) ou par les techniques de différentiation automatique (Gilbert, 1992) est soit très coûteuse, soit imprécise ou même impossible, étant donné les caractéristiques des fonctions de la boîte noire.

Les techniques d'OSD sont divisées en plusieurs catégories. On peut en citer les métaheuristiques (le recuit simulé (Kirkpatrick *et al.*, 1983), les algorithmes reposant sur une stratégie évolutionnnaire (Jebalia, 2008), les méthodes basées sur des opérations sur un simplexe (Nelder et Mead, 1965), les méthodes directes directionnelles (par motifs (Torczon, 1997), par treillis adaptifs (Audet et Dennis, Jr., 2006), multidirectionnelles (Dennis, Jr. et Torczon, 1991), utilisant les directions conjuguées (Rosenbrock, 1960) ainsi que les méthodes modélisant la fonction objectif (fonctions de substitution) soit par construction de surface de réponse (Jones *et al.*, 1998), soit en utilisant les fonctions

de base radiale (Björkman et Holmström, 2000) ou par interpolation dans une région de confiance (Powell, 2004), (Berghen, 2004), (Conn *et al.*, 1998), (Conn *et al.*, 2006), (Conn et Toint, 1996).
En l'occurence, nous nous intéressons particulièrement aux méthodes directes auxquelles nous consacrons la section suivante.

2.2 Méthodes de recherche directe

Les méthodes de recherche directe, connues aussi sous le nom de « méthodes d'ordre zéro » (vu qu'elles ne font pas de calcul de dérivées) (Lewis *et al.*, 2000), constituent une classe de l'OSD pour laquelle la retenue ou le rejet d'une solution courante repose uniquement sur des comparaisons algorithmiques des valeurs de la fonction objectif. Une caractérisation détaillée de ces méthodes est donnée dans (Trosset, 1997). Les méthodes directes ont surgi vers les années 1950 comme étant des heuristiques (dont les plus populaires sont : (Hooke et Jeeves, 1961) et (Nelder et Mead, 1965)) et ont été écartées vers le début des années 1970 en faveur de recherches fertiles sur les méthodes newtoniennes (Dennis, Jr. et Schnabel, 1983) . Elles ont connu un regain d'intérêt au début des années 1990 avec l'apparition des premiers résultats de convergence dans un contexte de programmation parallèle (Torczon, 1991). Dès lors, leur utilisation s'est avérée efficace à résoudre des problèmes d'optimisation complexes avec des propriétés de convergence rigoureuses.

2.3 Ensembles générateurs et bases positives

Les ensembles générateurs et les bases positives sont des notions importantes pour les algorithmes de recherche directe. Un ensemble générateur $[d_1 \dots d_l]$ est un ensemble de

vecteurs engendrant positivement l'espace $\mathbb{R}^n$:

$$\{v \in \mathbb{R}^n : v = \alpha_1 d_1 + \ldots + \alpha_l d_l, \alpha_i \geq 0 : i = 1, \ldots, l\} = \mathbb{R}^n.$$

La théorie des bases positives a été initialement introduite dans (Davis, 1954). Une base positive de $\mathbb{R}^n$ est un ensemble de vecteurs non-nuls indépendant (c-à-d de rang maximal pour lequel aucun vecteur ne peut s'écrire sous forme de combinaison linéaire des autres vecteurs) qui engendre $\mathbb{R}^n$ par des combinaisons linéraires positives (Conn *et al.*, 2009). La cardinalité d'une base positive varie entre $n + 1$ (dans ce cas elle est appelée base positive minimale) et $2n$ (base positive maximale). Le lecteur intéressé pourra se référer à (Audet, 2011) pour une preuve sur la cardinalité maximale d'une base positive. Sous forme matricielle, une base positive maximale de $\mathbb{R}^n$ correspond à :
$D\oplus = [I_n \ -I_n] = [e_1 \ldots e_n \ -e_1 \ldots -e_n]$, où $\{e_1, e_2, \ldots, e_n\}$ est la base canonique de $\mathbb{R}^n$.

La principale motivation sur laquelle repose l'utilisation des ensembles générateurs positifs pour certains algorithmes directionnels (Lewis et Torczon, 1996) est basée sur un théorème qui stipule qu'*un ensemble générateur* $[d_1 \ldots d_l]$ *engendre* $\mathbb{R}^n$ *positivement si et seulement si pour tout vecteur* $v \in \mathbb{R}^n$ *non-nul, il existe un indice dans* $\{1, \ldots, l\}$*, tel que* $v^\top d_i > 0$. En choisissant $v = -\nabla f(x)$ (lorsque $\nabla f(x)$ existe et est non-nul), alors $\nabla f(x)^\top d_i < 0$. Par conséquent, il existe au moins un indice $i \in \{1, \ldots, l\}$ tel que d_i est une direction de descente (formant un angle aigu avec v tel que le montre la figure 2.1).

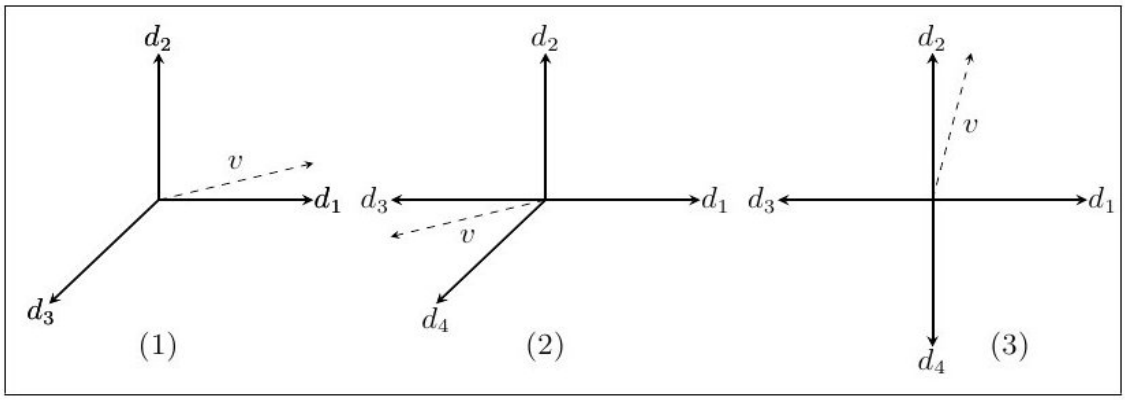

Figure 2.1 (1) est une base positive minimale de $\mathbb{R}^2$, (2) est un ensemble générateur positif mais pas une base positive, (3) est une base positive maximale. d_1 et d_2 sont des directions de descente pour (1) et (3), d_3 et d_4 le sont pour (2), lorsque $v = -\nabla f(x)$.

Dans les trois sections qui suivent, nous allons exposer les algorithmes CS, GPS et MADS. CS est l'ancêtre de GPS et en est un cas particulier. De même, l'algorithme MADS est une généralisation de GPS. Il est donc inévitable de passer au travers de l'un des deux, d'autant plus que nous souhaiterons comparer notre méthode à GPS et MADS à la suite de ce mémoire.

2.4 Algorithme de recherche par coordonnées (CS : *Coordinate Search* ou *Compass Search*)

La recherche par coordonnées (Fermi et Metropolis, 1952) est la méthode directe directionnelle la plus simple utilisant la base $D\oplus$ pour la génération de directions d'exploration.

On considère le cas non contraint pour lequel on s'intéresse à minimiser une fonction

objectif f sur $\mathbb{R}^n$ sans aucune contrainte :

$$\min_{x \in \mathbb{R}^n} f(x)$$

où : $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. On assigne la valeur infinie à f lorsque l'évaluation échoue. CS est une méthode itérative, on dénote par k le compteur du nombre d'itérations. On émet l'hypothèse qu'on peut fournir un point initial x_0 tel que $f(x_0) < \infty$ et on définit l'ensemble des points de la sonde (le *cadre*) par :

$$P_k = \{x_k + \Delta_k d : d \in D\oplus\}.$$

x_k : est l'itéré courant qui représente le meilleur point à date et est aussi appelé *centre de sonde*.

Δ_k : est le pas considéré à l'itération k qui sera désigné plus tard par *paramètre de taille du treillis*.

Algorithme 1 : L'algorithme de recherche par coordonnées

---

1 : **Initialisation :** $k \leftarrow 0$, $x_0 \in \mathbb{R}^n$ tel que $f(x_0) < \infty$ et $\Delta_0 > 0$
2 : **Sonde locale ou** POLL **:** Évaluer la fonction objectif aux points t de la sonde P_k. Si on trouve un point tel que $f(t) < f(x_k)$, alors : $x_{k+1} \leftarrow t$ et l'étape de sonde est déclarée comme réussie. Sinon, $x_{k+1} \leftarrow x_k$.
3 : **Mise à jour des paramètres :**
- Si l'étape de sonde est réussie, $\Delta_{k+1} \leftarrow \Delta_k$, sinon $\Delta_{k+1} \leftarrow \Delta_k/2$ et $x_{k+1} \leftarrow x_k$.
- $k \leftarrow k + 1$ et retourner à 2 si aucune condition de terminaison n'est satisfaite.

Les conditions de terminaison sont le plus souvent une tolérance sur la taille du treillis (c-à-d qu'on doit vérifier que $\Delta_k < \Delta_{tol}$) ou un nombre maximum d'évaluations de la fonction objectif.

Un algorithme CS de base consiste à évaluer $2n$ directions autour du centre de sonde courant (x_k). Si un point t appartenant à P_k est une amélioration de la solution courante, alors celui-ci devient le centre du cadre et le pas Δ_k demeure inchangé, sinon ce paramètre est réduit à la moitié.

D'autres stratégies pourront être utilisées pour optimiser CS dont une évaluation dynamique du voisinage de x_k commençant par une direction jugée prometteuse pour laquelle il y a eu un succès à une itération ultérieure ainsi que l'évaluation opportuniste des points d'essais qui arrêter la recherche locale lorsqu'un meilleur point est trouvé (on passe ainsi directement à l'étape 3 de l'algorithme 1).

Limites de CS **:**

Des exemples de la littérature ((Abramson, 2002), (Kolda *et al.*, 2003)) ont montré l'inefficacité de cette méthode. D'une part, elle considère un nombre limité de directions (toujours les mêmes $2n$ directions) ce qui limite considérablement l'espace de recherche, d'autre part elle est très sensible aux fonctions non lisses.

2.5 Algorithme de recherche par motifs (GPS : *Generalized Pattern Search*)

Les méthodes directes de recherche par motif ont été introduites par (Torczon, 1997) pour la résolution de problèmes de programmation non linéaire sans contraintes et étendues dans le cas des problèmes à contraintes de bornes (Lewis et Torczon, 1999), à contraintes générales linéaires (Lewis et Torczon, 2000) et non linéaires (Lewis et Torczon, 2002). La méthode généralisée de recherche par motif telle que décrite dans (Lewis et Torczon, 1996) génère, à chaque itération, une séquence finie d'itérés sur un maillage avoisinant la solution courante, construite à partir d'une combinaison linéaire positive de vecteurs d'un ensemble générateur positif. La fonction objectif est alors évaluée aux différents points de la séquence dans le but de trouver un itéré améliorant la solution actuelle. Si c'est le cas, alors le maillage est agrandi et l'itéré est retenu sinon le maillage est contracté et une nouvelle séquence est générée.

Nous présentons l'algorithme GPS de (Torczon, 1997) tel qu'il a été évoqué dans (Audet et Dennis, Jr., 2003). La gestion des contraintes sera abordée à la suite de ce chapitre. Le paramètre de la taille du treillis doit respecter la règle suivante :

$$\Delta_{k+1} = \tau^{\omega_k} \Delta_k, \tag{2.1}$$

où $\tau \in \mathbb{Q}$ et $\omega_k \in \mathbb{Z}$ compris entre $\omega^+ \geq 0$ et $\omega^- \leq -1$ tel que :

$$\omega_k \in \begin{cases} \{0, 1, \ldots, \omega^+\} & \text{si l'itération est réussie} \\ \{\omega^-, 1 + \omega^-, \ldots, -1\} & \text{sinon.} \end{cases} \tag{2.2}$$

Dans NOMAD, les valeurs par défaut sont : $\tau = 2$, $\omega^+ = 0$ et $\omega^- = -1$.
D représente l'ensemble des directions et doit être de la forme $D = GZ$ avec $G \in \mathbb{R}^{n \times n}$ une matrice non singulière et $Z \in \mathbb{Z}^{n \times n_D}$, selon (Audet, 2004).
$D_k \subseteq D$: représente un ensemble générateur positif de directions à l'itération k. Le treillis (ou le maillage) est une discrétisation spaciale de $\mathbb{R}^n$ incluant tous les points d'essais possibles et est défini par :

$$M_k = \left\{ x + \Delta_k D z : z \in \mathbb{N}^{|D|}, x \in V_k \right\}.$$

V_k est l'ensemble des points où la fonction objectif a été évaluée au début de l'itération k.

$$P_k = \{x_k + \Delta_k d : d \in D_k\} \subseteq M_k.$$

GPS introduit une étape de recherche qui n'apparaît pas dans CS : c'est la recherche globale ou SEARCH. La recherche globale est optionnelle et flexible et peut être opportuniste (un succès termine immédiatement cette étape) ou exhausive (l'ensemble des points de S_k est évalué). Elle permet d'exploiter les connaissances du problème en visant un nombre fini de points d'essai prometteurs. Elle consent à l'utilisation de différentes stratégies de recherche tels que les métaheuristiques de recherche à voisinage variable

(Audet *et al.*, 2008a), les modèles de substitution ((Booker *et al.*, 1999), (Conn et Le Digabel, 2011)) et l'échantillonnage par hypercube latin (Tang, 1993), afin de mieux explorer l'espace de recherche. L'incorporation de cette étape qualifie les méthodes directes directionnelles tels que MADS et GPS de méthodes hybrides qui associent des techniques autre que directionnelles afin de guider l'optimisation.

Algorithme 2 : L'algorithme de recherche par motif GPS

1 : **Initialisation** : $k \leftarrow 0$, $x_0 \in \Omega$ tel que $f(x_0) < \infty$, $\Delta_0 > 0$, τ, ω^- et ω^+.

2 : **Recherche globale ou** SEARCH : Évaluer f sur S_k un sous ensemble fini de M_k en utilisant une stratégie prédéterminée afin de trouver $t \in S_k$ tel que $f(t) < f(x_k)$.

3 : **Sonde locale ou** POLL : Si la recherche globale n'est pas un succès, considérer $D_k \subseteq D$ et évaluer f aux points de $P_k \subset M_k$.

4 : **Mise à jour des paramètres :**

- S'il existe $t \in T_k = S_k \bigcup P_k$ tel que $f(t) < f(x_k)$ (itération réussie) alors $x_{k+1} \leftarrow t$, sinon $x_{k+1} \leftarrow x_k$ (itération échouée).
- Mettre à jour ω_k selon (2.2) et Δ_{k+1} selon (2.1).
- $k \leftarrow k + 1$ et retourner à 2 si aucune condition de terminaison n'est satisfaite.

À l'itération k, l'ensemble des points d'essai définit T_k. Si $f(t) < f(x_k)$, pour tout $t \in T_k$, alors y est *un point améliorant du treillis*. Si $f(x_k) \leq f(y)$, pour tous les $y \in P_k$, alors x_k est un *optimum local du treillis*.

On peut remarquer que l'algorithme CS est bel et bien un cas particulier de GPS, si on prend $D = D\oplus, \omega^+ = 0, \omega^- = -1$ et $\tau = 2$ pour l'algorithme 2 en excluant la recherche globale ($T_k = P_k$).

Limites de GPS **:**

On retrouve des exemples pathologiques pour GPS dans la littérature qui sont dûs à un nombre fini de directions de sonde qui mène soit à de faibles résultats de convergence (Audet, 2004), soit à la non-optimalité des résultats (Kolda *et al.*, 2003).

2.6 Algorithme de recherche par treillis adaptifs (MADS : *Mesh Adaptive Direct Search)*

L'algorithme MADS constitue une généralisation de GPS dont une première implémentation est proposée dans (Audet et Dennis, Jr., 2006) avec différentes possibilités de choix de directions. Il est destiné à améliorer GPS en offrant une exploration plus efficace de l'espace et des résultats de convergence robustes. Ainsi, il permet d'en combler les lacunes observées dans (Audet, 2004). Nous nous situons désormais dans le cadre des problèmes contraints (où les contraintes peuvent être non linéaires). Plus précisément, nous nous intéressons à ceux de la forme :

$$\min_{x \in \Omega} f(x)$$

où $\Omega = \{x \in X : g_j(x) \leq 0, j \in J = \{1, 2, \ldots, m\}\}$, $f, g_j : X \to \mathbb{R} \cup \{\infty\}$, X et Ω étant des sous-ensembles de $\mathbb{R}^n$.

Ω est le domaine réalisable pour lequel on ne pose aucune hypothèse de linéarité, de convexité, etc. Il est défini à travers X qui représente l'ensemble des contraintes *non relaxables* ou inviolables. f et g_j sont des fonctions de boîtes noires qui portent les caractéristiques mentionnées en introduction. Les g_j sont des contraintes *relaxables* ou violables qui procurent une mesure de la violation subie. D'autres contraintes qualifiées

de *cachées* font en sorte que la boîte noire peut échouer à retourner une valeur (par exemple dans le cas où la résolution d'un système d'équations différentielles n'a aucune solution ou simplement pour des raisons inconnues propres à la boîte noire) ce qui se traduit formellement par une valeur de retour infinie.

À la différence de GPS qui ne considère qu'un seul paramètre Δ_k du treillis, l'algorithme MADS apporte une légère modification à celui-ci, il devient alors Δ_k^m afin de le distinguer du paramètre de *taille de cadre* Δ_k^p.

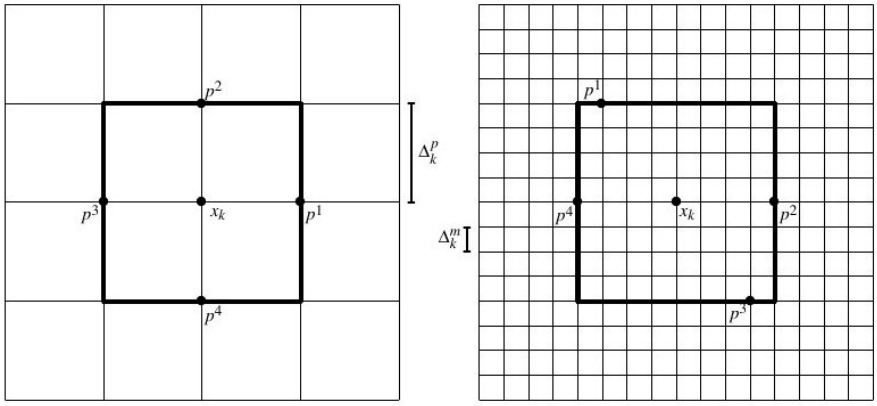

Figure 2.2 Les cadres de GPS (gauche) et MADS (droite) en gras, $P_k = \{p_1, p_2, p_3, p_4\}$, figure tirée de (Abramson et Audet, 2006)

La figure 2.2 illustre la distinction entre GPS et MADS ainsi que le rôle de Δ_k^m et Δ_k^p. Dans la partie gauche de la figure, la taille du cadre Δ_k^p est la même pour GPS et MADS. Pour MADS, la taille du treillis Δ_k^m est beaucoup plus faible que la taille du cadre Δ_k^p. Par conséquent, il présente plusieurs possibilités de constructions de points de sonde, ce qui constitue sa contribution majeure à générer un ensemble de directions dense.

Compte tenu du paramètre Δ_k^m, les expressions de M_k et P_k deviennent :

$$M_k = \left\{ x_k + \Delta_k^m Dz : z \in \mathbb{N}^{|D|}, x \in V_k \right\},$$

$$P_k = \{x_k + \Delta_k^m d : d \in D_k\} \subseteq M_k.$$

La distinction entre GPS et MADS est que dans MADS, D_k n'est pas contraint à être un sous-ensemble de D.

Pour tout $d \in D_k$ ($d \neq 0$), une direction de sonde :

– Il existe $u \in \mathbb{N}^{|D_k|}$ tel que $d = Du$;

– La distance entre le centre du cadre x_k et un point du cadre $t \in P_k$ est bornée par un multiple de Δ_k^p : $\Delta_k^m \|d\| \leq \Delta_k^p \max\{\|d'\| : d' \in D\}$;

– Les limites des ensembles D_k normalisés ($D_k = \left\{\frac{d}{\|d\|} : d \in D_k\right\}$), tels que définis dans (Price et Coope, 2003), sont des ensembles générateurs positifs.

Les paramètres Δ_k^m et Δ_k^p doivent vérifier les conditions suivantes :

$$\Delta_k^m \leq \Delta_k^p, \text{ pour tout } k, \tag{2.3}$$

et

$$\lim_{k \in K} \Delta_k^m = 0 \Leftrightarrow \lim_{k \in K} \Delta_k^p = 0, \text{ pour tout ensemble fini d'indices K.} \tag{2.4}$$

Mise à part les différences mentionnées, un algorithme MADS correspond à l'algorithme 2 de la section précédente.

Gestion des contraintes :

Chaque contrainte est gérée en utilisant l'une des trois approches suivantes : la barrière extrême (EB pour *Extreme Barrier*), la barrière progressive (PB pour *Progressive Barrier*) et la barrière progressive à extrême (PEB pour *Progressive to Extreme Barrier*) qui

est une combinaison des deux approches précédentes .

La barrière extrême (appliquée sur MADS dans (Audet et Dennis, Jr., 2006)) consiste à considérer la fonction f_Ω suivante qui rejète automatiquement tous les points d'essai t qui ne font pas partie du domaine réalisable ce qui revient à considérer le problème sans contraintes ($\Omega = \mathbb{R}^n$) :

$$f_\Omega(t) = \begin{cases} f(t) & \text{si } t \in \Omega, \\ \infty & \text{sinon.} \end{cases}$$

Cette approche peut permettre de réduire le nombre d'évaluations de la fonction objectif (qui est coûteuse) en excluant l'ensemble des solutions non réalisables.

La barrière progressive est issue de la méthode du filtre (Fletcher et Leyffer, 2002) qui a été utilisée avec GPS (Audet et Dennis, Jr., 2004) pour les problèmes à contraintes générales et adaptée pour MADS dans (Audet et Dennis, Jr., 2009). Elle consiste à introduire une fonction non négative h qui mesure la violation des contraintes :

$$h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\} .$$

$$h(t) := \begin{cases} \infty & \text{si } t \notin X, \\ \sum_{j \in J} (\max(g_j(t), 0))^2 & \text{sinon.} \end{cases}$$

Il résulte de la définition de h que :

– $t \in \Omega \Leftrightarrow h(t) = 0$;

– Si $0 < h(t) < \infty$ alors $t \in X \backslash \Omega$.

Cette approche tolère les points d'essais qui violent les contraintes relaxables. Plus précisément, à chaque itération k, les points d'essai dont la mesure de la violation n'excède pas un certain seuil $h_k^{\max}$ (qui décroît tout au long de l'itération) sont évalués. Les autres points sont simplement écartés. Le problème d'optimisation contraint est ainsi perçu comme un problème biobjectif minimisant à la fois f et h avec une priorité ac-

cordée à la réduction de h, afin d'obtenir des solutions réalisables.

La barrière progressive à extrême (Audet *et al.*, 2010b) traite initialement les contraintes relaxables par la barrière progressive et passe à la barrière extrême dès que ces contraintes deviennent satisfaites. En effet, si le sondage autour d'un centre de cadre non réalisable génère un point vérifiant une contrainte violée par le centre de sonde locale, alors celle-ci est désormais traitée avec la barrière extrême.

Un avantage important de PB et PEB est qu'ils ne requièrent pas de point initial réalisable, ce qui est intéressant de point de vue pratique pour les problèmes industriels, où il n'est pas toujours faisable d'en fournir un.

Convergence de MADS **:**

L'analyse convergence de MADS ainsi que toutes ses instanciations repose sur le calcul non lisse de Clarke (Clarke, 1983) et est présentée de manière hiérarchique, dépendamment des hypothèses posées sur f et Ω et la façon avec laquelle on gère les contraintes. Ainsi, pour chaque stratégie EB, PB et PEB, on retrouve une analyse de convergence respectivement dans (Audet et Dennis, Jr., 2006), (Audet et Dennis, Jr., 2009) et (Audet *et al.*, 2010b). Les définitions sur lesquelles les résultats suivants reposent sont en annexe.

On se place dans le cas général d'un problème contraint. On suppose que l'on peut fournir un point initial dans X mais pas nécessairement dans Ω et que tous les itérés appartiennent à un ensemble compact.

Les conditions (2.3) et (2.4) impliquent le résulat suivant et assurent la convergence de MADS :

$$\lim_{k\to+\infty} \inf \Delta_k^p = \lim_{k\to+\infty} \inf \Delta_k^m = 0.$$

(Audet et Dennis, Jr., 2003) ont prouvé l'existence d'une sous-suite raffinante (*définition I.2*) d'optima locaux $\{x_k\}_{k\in K}$ qui converge. Notons $\hat{x}$ le point limite de cette sous-suite.

Compte tenu de la gestion des contraintes, $\hat{x}$ peut être la limite d'une sous-suite de centres de cadres réalisables ou non.

Dans le premier cas, il a été montré dans les articles cités que :

– Si aucune hypothèse n'est émise sur f, alors $\hat{x}$ est le point limite d'une sous-suite d'optima locaux devenant infiniment fins. De plus, si f est semi-continue inférieurement près de $\hat{x}$, alors $f(\hat{x}) \leq \lim_{k} f(x_k)$;
– Si f est Lipschitz près de $\hat{x}$ (***définition I.1***) alors les dérivées généralisées de Clarke (***définition I.3***) $f^{\circ}(\hat{x}, d) \geq 0, \forall d \in T_{\Omega}^{H}(\hat{x})$ (***définition I.4***). Si $T_{\Omega}^{H}(\hat{x})$ est non vide, alors $f^{\circ}(\hat{x}, d) \geq 0, \forall d \in T_{\Omega}^{Cl}(\hat{x})$ (***définition I.5***), c-à-d $\hat{x}$ est un point Clarke-stationnaire ;
– Si f et Ω sont réguliers près de $\hat{x}$ (***définition I.7***), alors $f^{\circ}(\hat{x}, d) \geq 0, \forall d \in T_{\Omega}^{B}(\hat{x})$ (***définition I.6***), c'est à dire, $\hat{x}$ est Bouligand-stationnaire ;
– Si f est strictement différenciable près de $\hat{x}$ et Ω régulier, alors $\hat{x}$ est Bouligand-KKT-stationnaire.

Sous de plus amples hypothèses, (Abramson et Audet, 2006) ont établi des résultats de convergence de deuxième ordre : si f est deux fois strictement différentiable, Ω est convexe et $\nabla^2 f(\hat{x})$ est non singulier, alors $\hat{x}$ est un minimum local strict.

Dans le deuxième cas (le cas où les itérés sont non réalisables), les résultats de convergence tiennent pour le problème suivant :

$$\min_{x \in X} h(x)$$

Dans le cas où $\hat{x} \in \Omega$, sous l'hypothèse supplémentaire suivante selon laquelle on peut garantir qu'il n'y a pas de direction de descente dans le cône hypertangent :

$\forall d \in T_{\Omega}^{H}(\hat{x}) \neq \emptyset$, il existe $\epsilon > 0$ pour lequel $h^{\circ}(x; d) < 0$ pour tout $x \in X \cap B_{\epsilon}(\hat{x})$ qui vérifie $h(x) > 0$, où $B_{\epsilon}(\hat{x})$ est la boule de centre $\hat{x}$ et de rayon ϵ.

2.7 De LTMADS à ORTHOMADS

LTMADS est la première instance de MADS proposée dans (Audet et Dennis, Jr., 2006) et en est une implémentation stochastique. L'appellation LTMADS dérive de *lower triangular matrix* où le choix des directions se fait à partir d'une matrice triangulaire inférieure dont on permute aléatoirement les lignes et les colonnes afin de diversifier les résultats.

Les paramètres de cette première implémentation correspondent à :

$$D = Z = [I_n \quad -I_n], \tau = 4, \omega^+ = 1, \omega^- = -1, \Delta_0^m = \Delta_0^p = 1.$$

Deux possibilités de complétion d'une base positive sont proposées : $n+1$ et $2n$ directions de sonde.

L'ensemble des directions de sonde normalisées générées par LTMADS est dense dans la boule unité avec une probabilité 1. Cela garantit sous certaines hypothèses la convergence de cette instance de MADS (Audet *et al.*, 2008b) ce qui implique sa validité.

ORTHOMADS est une instance de MADS introduite dans (Abramson *et al.*, 2009) dont l'émergence repose sur deux principaux motifs : le premier étant la possibilité de reproduire les résultats d'une expérience, étant donné que l'algorithme utilise la suite quasi-aléatoire de Halton (Halton, 1960) pour générer des directions déterministes orthogonales, le deuxième est lié au fait que ORTHOMADS permet d'éviter les angles assez grands entre les directions de sonde à une itération donnée (donc les régions non explorées), ce qui constitue un inconvénient de LTMADS. Des tests provenant de problèmes tirés de la littérature ont montré la dominance de cette instance sur la précédente pour une seule utilisation (ce qui correspond au cas des boîtes noires où l'on réduit le nombre d'appels). ORTHOMADS est déterministe, cependant un paramètre de la suite de Halton permet de varier les résultats à chaque utilisation. Ce paramètre correspond au $n^{\text{ème}}$ nombre premier pour le logiciel NOMAD, où n est la dimension du problème.

Nous gardons un intérêt particulier pour cette instance et nous y serons de retour dans un chapitre prochain, étant donné que c'est l'instance par défaut de NOMAD qu'on utilisera pour développer notre algorithme. Notons que le même résultat de convergence que LTMADS est assuré, sans terme de probabilité, puisque ORTHOMADS est déterministe.

CHAPITRE 3

REVUE DE LA LITTÉRATURE : MÉTHODES D'ANALYSE DE SENSIBILITÉ

Dans le présent chapitre, nous donnons un aperçu sur les méthodes d'analyse de sensibilité (AS) dont l'objectif principal concorde avec le nôtre. Nous tâchons de trouver, parmi celles-ci, une méthode conforme aux spécifications mentionnées au chapitre 1. Nous introduisons aussi les concepts statistiques en analyse de la variance (ANOVA) qui nous seront utiles au chapitre suivant.

3.1 Introduction à l'analyse de sensibilité

L'étude de l'incidence des variables d'entrée sur la variable de sortie pour l'identification des variables influentes nous amène à l'AS de la sortie par rapport à chacune des entrées. L'AS peut être définie comme étant l'étude de l'impact de la variation des entrées sur la variation de la sortie.

On considère un modèle mathématique de simulation ou de prédiction décrivant un processus donné (chimique, physique, biologique, financier, etc.) et retournant une sortie Y qu'on supposera unidimensionnelle :

$$Y = f(X_1, X_2, \ldots, X_k). \tag{3.1}$$

Sauf indication contraire, pour la suite de ce chapitre, $X_1, X_2, ..X_k$ sont des variables indépendantes et k est le nombre de variables d'entrée.

La fonction f peut être très complexe, lorsque par exemple, son évaluation requiert la résolution d'un système d'équations différentielles. En pratique, f est calculée par un code informatique.

Selon (Saltelli *et al.*, 2000) l'AS permet de :

→ Déterminer si le modèle décrit bien le processus qu'il représente ;

→ Identifier les facteurs qui contribuent à la variabilité de la sortie et qui requièrent plus d'intérêt afin d'améliorer le modèle ;

→ Déterminer les paramètres non significatifs qui peuvent être éliminés du modèle ;

→ Détecter s'il y a des interactions entre certaines variables d'entrée ou un groupe de variables ;

Dans le cas où on ne peut pas émettre d'hypothèses sur le modèle et où la modélisation du processus n'est pas l'objectif visé de l'AS, (Saltelli *et al.*, 2004) rapportent d'autres éléments qui correspondent mieux à ce contexte :

→ Hiérarchisation des facteurs (*Factors Prioritisation*) : identification du facteur le plus influent qui prend la grande part de variance de la sortie ainsi que ceux qui en prennent de moins en moins, ce qui permet de classer les facteurs ;

→ Fixation de facteurs (*Factors Fixing*) : identification du facteur ou du groupe de facteurs qu'on peut fixer, car ils n'influencent pas la variance de la sortie ;

→ Réduction de la variance (*Variance Cutting*) : réduction de la variance au dessous d'un seuil donné en fixant simultanément le moins de facteurs possible. Ceci est utile surtout en analyse de risque ;

→ Cartographie de la sortie (*Factors Mapping*) : lorsque la sortie Y est répartie selon des régions de différentes caractéristiques (par exemple acceptable ou non), alors on détermine le facteur qui est à l'origine de cette répartition.

On s'intéresse, dans cette revue de la littérature, aux méthodes qualitatives et quantitatives d'AS qui tentent de déterminer les variables influentes ou de hiérarchiser les variables d'entrée en fonction de leur importance sur la sortie. Par conséquent, les mesures de sensibilité qui permettent d'étudier la relation entre l'entrée et la sortie (linéarité, monotonie, etc.) tel que le coefficient de détermination ou de corrélation (de Pearson) ne feront pas l'objet de notre étude. Pour cela, on se limitera aux méthodes locales, aux méthodes de criblage et aux méthodes globales basées sur la décomposition de la va-

riance et la régression linéaire.

3.2 Méthodes locales

Les méthodes locales donnent, tel que leur nom l'indique, une information sur l'impact de la variation d'une variable d'entrée à un niveau local, c'est à dire, lorsque celle-ci prend une valeur nominale donnée x_0. Cela revient à évaluer les k dérivées partielles $\frac{\delta Y}{\delta X_i}|_{X=x_0}$ du modèle (3.1) ou à les estimer en utilisant, par exemple, la dérivation automatique de codes. Ces quantitées sont des coefficients de sensibilité (Saltelli *et al.*, 2000) représentant une estimation linéaire du nombre d'unités de variation de Y suite à une variation d'une unité de X_i.
Afin d'être indépendant de l'unité de grandeur, ces coefficients peuvent être normalisés en considérant la variance des X_i et celle de Y :

$$\hat{S}_i = \frac{V(X_i)}{V(Y)} \frac{\delta Y}{\delta X_i}|_{X=x_0}.$$

Le classement des variables se fait alors en fonction des coefficients normalisés. Les facteurs les plus influents ont un coefficient plus élevé. La principale limite de ces méthodes est le caractère local de l'étude. Le classement de variables obtenu est convenable seulement si le modèle est linéaire ou quasi-linéaire.

3.3 Méthodes de criblage

Les méthodes de criblage (*screening*) sont des méthodes économiques en temps de calcul qui visent à identifier qualitativement les variables d'entrées importantes pour les modèles de grande dimension (des centaines d'entrées) difficiles à explorer. Dans la suite de cette section, on se limitera aux méthodes de criblage permettant l'identification

d'un petit nombre de variables d'entrée d'un modèle. On y retrouve les designs OAT (*One At a Time*), la méthode de Morris (Morris, 1991), les designs de (Cotter, 1979) et de (Andres et Hajas, 1993) et bien d'autres.

Les designs OAT sont des méthodes de criblage typiques et simples qui consistent à étudier l'influence de la variation d'une variable d'entrée donnée à la fois. Chaque variable prend deux ou trois modalités. À deux modalités, $k + 1$ évaluations sont requises (Iooss, 2011). Quoique les méthodes OAT soient locales et ne tiennent pas compte des interactions, (Morris, 1991) a proposé une méthode globale, couvrant tout l'espace des variables d'entrées, qui repose sur un design OAT permettant de déterminer les facteurs non influents ou les facteurs ayant un effet linéaire ou additif ou bien ceux dont l'effet est non linéaire ou avec interactions. L'idée de base de cette méthode est une analyse statistique d'un échantillon d'effets élémentaires (dérivées partielles) des variables afin de mesurer l'importance relative des entrées. Cela consiste à calculer plusieurs réalisations d_i d'un facteur à la fois :

$$d_i = \frac{1}{\Delta}(f(x_1, \ldots, x_{i-1}, x_i + \Delta, x_{i+1}, \ldots, x_k) - f(x_1, \ldots, x_k)),$$

où $\Delta \neq 0 \in \mathbb{R}$.

Une moyenne élevée des d_i indique que le facteur x_i a une grande influence sur la variabilité de la sortie, tandis qu'une variance élevée renseigne sur une non-linéarité ou une corrélation avec les autres facteurs, sans pour autant pouvoir le trancher.

(Cotter, 1979) a présenté un design systématique factoriel répliqué nécessitant $2k + 2$ évaluations. Cependant, celui-ci n'est pas assez précis et ne détecte pas les facteurs importants dont les effets s'annulent mutuellement. (Andres et Hajas, 1993) ont développé un design factoriel fractionnaire itératif permettant d'estimer les effets principaux et les effets quadratiques avec un nombre faible d'évaluations (par rapport au nombre de facteurs), en faisant des regroupements de facteurs. Une autre technique de criblage par groupe a été proposée par (Bettonvil et Kleijnen, 1997). Enfin, (Saltelli *et al.*, 2008) ont

décrit une méthode de criblage (*Elementary Effets method*) ayant la flexibilité d'un design OAT qui permet de surmonter quelques limites mentionnées et peut être étendue dans le cas de groupement de facteurs, lorsque cela est nécessaire.

3.4 Méthodes globales

Les méthodes globales tiennent compte de tout l'espace de variation de la sortie afin de quantifier la part de la variance relative à chacune des entrées. Deux propriétés permettent de caractériser les méthodes globales selon (Saltelli *et al.*, 2000) : la forme et le domaine de la densité de probabilité de la fonction de sortie ainsi que l'étude de la sensibilité d'une variable X_i en faisant varier toutes les autres variables. Nous décrivons ici quelques méthodes basées sur la décompostion de la variance de la sortie et la régression linéaire. D'autres méthodes basées, entres autres, sur les tests statistiques et les métamodèles sont exposées dans (Iooss, 2011).

3.4.1 Méthodes basées sur la variance

L'intérêt des méthodes d'AS basées sur la variance réside dans leur indépendance du modèle et leur capacité de capturer la variance des facteurs d'entrée ainsi que les effets d'interactions. Le seul inconvénient est le coût de l'estimation. Parmi les méthodes d'AS basées sur la variance, les méthodes de Sobol et FAST sont les plus connues.

Méthode de Sobol

(Sobol, 2001) a établi, sous certaines hypothèses, à partir d'une décomposition connue sous le nom de HDMR (pour *High Dimensional Model Representation*), les mesures de

sensibilité suivantes :

$$S_{i_1 \ldots i_s} = \frac{D_{i_1 \ldots i_s}}{D}, \qquad 1 \leq i_1 < \ldots < i_s \leq k, \tag{3.2}$$

où

$$D = \sum_{s=1}^{k} \sum_{i_1 < \ldots < i_s}^{k} D_{i_1 \ldots i_s}, \tag{3.3}$$

D et $D_{i_1 \ldots i_s}$ sont des constantes appelées variances.

Les S_i sont appelés indices de sensibilité d'ordre un de Sobol correspondant au facteur X_i et permettant de quantifier son effet (effet principal) sur la sortie.
De même, l'indice de sensibilité d'ordre deux S_{ij} permet de quantifier l'effet de l'interaction entre les facteurs X_i et X_j et ainsi de suite. À partir de (3.3), on peut déduire que la somme de tous les indices de tous les ordres est égale à 1 :

$$\sum_{s=1}^{k} \sum_{i_1 < \ldots < i_s}^{k} S_{i_1 \ldots i_s} = 1.$$

On définit également l'indice de sensibilité total du facteur X_i par :

$$S_{T_i} = 1 - S_{-i}, \tag{3.4}$$

où S_{-i} est la somme de tous les termes $S_{i_1 \ldots i_s}$ n'incluant pas l'indice i.

Théorème de la variance conditionnelle :

La variance de la sortie $V(Y)$ peut être aussi décomposée en utilisant les variances conditionnelles à X_j et X_{-j} (X_{-j} désigne tous facteurs sauf celui d'indice j). Cette

décomposition est valable dans tous les cas, indépendemment de l'orthogonalité des entrées (Saltelli *et al.*, 2004) :

$$V(Y) = V(E(Y|X_j)) + E(V(Y|X_j)),$$

et

$$V(Y) = V(E(Y|X_{-j})) + E(V(Y|X_{-j})),$$

où $V(E(Y|X_j))$ est la variance de l'espérance de Y conditionnellement à X_j, et $E(V(Y|X_j))$ est l'espérance de la variance Y conditionnellement à X_j.
En normalisant les deux décompositions, on obtient :

$$1 = \frac{V(E(Y|X_j))}{V(Y)} + \frac{E(V(Y|X_j))}{V(Y)},$$

et

$$1 = \frac{V(E(Y|X_{-j}))}{V(Y)} + \frac{E(V(Y|X_{-j}))}{V(Y)}.$$

Il est alors démontré dans (Saltelli *et al.*, 2000) que l'indice de sensibilité du premier ordre de Sobol, ainsi que celui d'ordre total, traduisant tous les deux l'influence de X_i sur Y, peuvent s'écrire sous la forme :

$$S_i = \frac{V(E(Y|X_i))}{V(Y)} = \frac{D_i}{D}, \tag{3.5}$$

$$S_{T_i} = \frac{V(E(Y|X_{-i}))}{V(Y)} = \frac{D_{-i}}{D}. \tag{3.6}$$

Il est également démontré dans (Saltelli *et al.*, 2000) que l'indice de sensibilité d'ordre

deux traduisant l'influence de deux variables X_i et X_j sur Y peut s'écrire sous la forme :

$$S_{ij} = \frac{V(E(Y|X_i, X_j)) - V(E(Y|X_i)) - V(E(Y|X_j))}{V(Y)} = \frac{D_{ij}}{D}. \tag{3.7}$$

Les indices d'ordre supérieur sont définis de façon similaire.

Estimation des indices de Sobol :

Nous présentons une méthode d'estimation des indices de sensibilité de Sobol attribuable à (Saltelli, 2002) qui constitue une amélioration de la méthode originale de (Sobol, 1990) qui requiert $N \times 2^k$ évaluations, étant donné le modèle (3.1) et un échantillon de taille N.

L'indice de sensibilité d'ordre un est estimé par :

$$\hat{S}_j = \frac{\hat{U}_j - \hat{E}^2(Y)}{V(Y)}.$$

L'évaluation de cette quantité nécessite la génération de deux échantillons de matrice de même taille :

$$M_1 = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_k^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_k^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_k^{(N)} \end{pmatrix},$$

et

$$M_2 = \begin{pmatrix} x_1^{(1')} & x_2^{(1')} & \cdots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \cdots & x_k^{(2')} \\ \vdots & \vdots & & \vdots \\ x_1^{(N')} & x_2^{(N')} & \cdots & x_k^{(N')} \end{pmatrix}.$$

La moyenne des observations $E(Y)$ est calculée à partir de M_1 ou M_2 (ou les deux), tandis que, pour estimer U_j, on définit une troisième matrice N_j obtenue à partir de M_1, où toutes les colonnes sauf celle de la variable X_j sont ré-échantillonnées.

$$N_j = \begin{pmatrix} x_1^{(1')} & x_2^{(1')} & \cdots & x_j^{(1)} & \cdots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \cdots & x_j^{(2)} & \cdots & x_k^{(2')} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_1^{(N')} & x_{21}^{(N')} & \cdots & x_j^{(N)} & \cdots & x_k^{(N')} \end{pmatrix}.$$

Les estimations de $E(Y)$ et U_j sont données par :

$$\hat{E}(Y) = \frac{1}{N}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}),$$

$$\hat{U}_j = \frac{1}{N-1}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) f(x_1^{(r')}, x_2^{(r')}, \ldots, x_{(j-1)}^{(r')}, x_j^{(r)}, x_{(j+1)}^{(r')}, \ldots, x_k^{(r')}).$$

Le coût associé permettant l'estimation des indices de sensibilité de premier ordre est de $N \times (k+1)$ évaluations de f.

Enfin, l'indice de sensibilité total est estimé par :

$$\hat{S}_{Tj} = 1 - \frac{\hat{U}_{-j} - \hat{E}^2(Y)}{\hat{V}(Y)},$$

où :

$$\hat{U}_{-j} = \frac{1}{N-1}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) f(x_1^{(r)}, x_2^{(r)}, \ldots, x_{(j-1)}^{(r)}, x_j^{(r')}, x_{(j+1)}^{(r)}, \ldots, x_k^{(r)}).$$

Un coût supplémentaire de Nk est dû à l'estimation des indices totaux. Une extension de cette méthode permet d'obtenir tous les indices d'ordre un, d'ordre deux et totaux au coût de $N \times (k+2)$ évaluations.

Méthode de FAST

FAST, l'acronyme de *Fourier Amplitude Sensitivity Test*, est une méthode qui a été développée par (Cukier *et al.*, 1973), (Cukier *et al.*, 1978), (Cukier *et al.*, 1975) et (Schaibly et Shuler, 1973) dans un contexte d'AS et d'incertitude.

Définissons d'abord le moment d'ordre r de Y du modèle (3.1) (Chan *et al.*, 1997) :

$$E(Y^r) = \int_{\Omega^k} f^r(x_1, x_2, \ldots, x_k) p(x_1, x_2, \ldots, x_k) dx. \tag{3.8}$$

où Ω^k est l'espace de variation des k variables d'entrée et p est la densité de probabilité de $X = (X_1, \ldots, X_k)$.

La méthode de FAST repose principalement sur la conversion de l'intégrale k-dimensionnelle en x (3.8) en une intégrale unidimensionnelle en s en utilisant la transformation suivante : $x_i = G_i(\sin(\omega_i))$, pour $i = 1, 2, \ldots, k$.

En utilisant les propriétés des séries de Fourier, la variance de la sortie peut être approchée par :

$$\hat{V}(Y) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f^2(s) ds - [E(Y)]^2.$$

où : $f(s) = f(G_1(\sin(\omega_1 s)), \ldots, G_k(\sin(\omega_k s)))$ et $E(Y) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)ds$.
Un bon choix de G_i serait, par exemple : $x_i = \frac{1}{2} + \frac{1}{\pi}\arcsin(\sin\omega_i s)$(Saltelli *et al.*, 2000).
Rappelons que les coefficients de Fourier sont définis par :

$$A_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\cos(js)ds,$$

$$B_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\sin(js)ds.$$

Une approximation de $V(Y)$ serait :

$$\hat{V}(Y) \simeq \sum_{j=-\infty}^{\infty}(A_j^2 + B_j^2) - (A_0^2 + B_0^2) \simeq 2\sum_{j=1}^{k}(A_j^2 + B_j^2).$$

Alors, la contribution du facteur X_i à la sortie Y peut être approchée par :

$$\hat{V}_{\omega_i}(Y) = 2\sum_{j=1}^{M}(A_{j\omega_i}^2 + B_{j\omega_i}^2),$$

où M est l'harmonique maximal, en général 4 ou 6.
Les indices de sensibilité de FAST sont donc estimés par :

$$\hat{S}_i = \frac{\hat{V}_{\omega_i}(Y)}{\hat{V}(Y)} = \frac{\sum_{j=1}^{M}(A_{j\omega_i}^2 + B_{j\omega_i}^2)}{\sum_{j=1}^{k}(A_j^2 + B_j^2)}.$$

Afin de pouvoir classer les variables d'entrée en utilisant la méthode de FAST, il faut définir ω_i et G_i et calculer A_j et B_j en évaluant la fonction modèle en un nombre suffisant de points.

3.4.2 Méthodes basées sur la régression linéaire multiple

Si le modèle est linéaire, les coefficients de régression peuvent servir d'estimateurs quantitatifs de sensibilité. Un modèle de régression linéaire est donné par :

$$Y_i = \beta_0 + \sum_{j=1}^{k} \beta_j X_{ij} + \epsilon_i, \qquad i = 1, 2, \ldots, N. \tag{3.9}$$

X_{ij} représente l'observation i de la variable X_j.

Les X_{ij} sont des données obtenues à l'aide d'une stratégie d'échantillonnage (échantillon de dimension N).
Les Y_i représentent les sorties correspondantes aux échantillons $(X_{i1}, \ldots, X_{ij}, \ldots, X_{ik})$, $i \in \{1, \ldots, N\}$. Les β_j sont les coefficients de régression, des constantes à déterminer par la méthode des moindres carrés et les ϵ_i sont des termes d'erreur.
Sous l'hypothèse d'indépendance des variables d'entrées X_j et en supposant aussi que celles-ci sont indépendantes des ϵ_i, la variance du modèle (3.9) peut être décomposée comme suit :

$$V = \sum_{j=1}^{k} V_j + V_e.$$

V_j la part de la variance expliquée par la variable X_j et la variance totale V sont estimées dans (Xu et Gertner, 2008) par :

$$\hat{V}_j = \hat{\beta}_j^2 V(X_j) = \frac{1}{N-1} \hat{\beta}_j^2 \sum_{i=1}^{N} (X_{ij} - \bar{X}_j)^2$$

$$\hat{V} = V(Y) = \frac{1}{N-1} \sum_{i=1}^{N} (Y_i - \bar{Y})^2.$$

La sensibilité relative à la variable X_j est alors donnée par :

$$S_j = \frac{\hat{V}_j}{\hat{V}}.$$

Cet indice de sensibilité est connu sous le nom de SRC pour (*Standardized Regression Coefficient*). Il est compris entre 0 et 1 et permet le classement de variables.
Si le modèle n'est pas linéaire mais monotone, alors un autre indicateur peut être utilisé. Il s'agit de l'indicateur SRRC pour *Standardized Rank Regression Coefficient*, basé sur la transformation des rangs. Cette dernière consiste à attribuer la valeur 1 à la plus petite valeur de sortie, 2 à celle qui lui est directement supérieure et ainsi de suite jusqu'à attribuer N à la plus grande valeur. On applique le même principe avec les k variables d'entrée. Le SRRC se calcule, alors, de la même façon que SRC, en considérant les vecteurs rangs. Notons que les β_j correspondent dans ce cas à β_j^R : coefficients de régression associés aux rangs.
Le SRRC a été utilisé, en guise d'exemple, dans (Allard *et al.*, 2011), avec deux autres méthodes (FAST et polynômes locaux (Da-Veiga, 2005)) pour l'identification des grandeurs qui influencent les paramètres d'un modèle de simulation incendie. Le classement des variables importantes trouvé est le même pour ces trois méthodes. D'autres indicateurs de sensibilité sont employés avec différentes techniques d'échantillonnage sur des problèmes tests linéaires, monotones et non monotones dans (Helton et Davis, 2002).

Nous n'allons pas utiliser directement une des méthodes d'AS évoquées dans le cadre de cette revue de la littérature pour des raisons que nous expliquerons au chapitre 4. Celui-ci présentera une méthode basée sur la décomposition de l'ANOVA qui sera en mesure d'estimer les indices d'ordre un et d'ordre supérieur que nous avons définis plus tôt.

CHAPITRE 4

MÉTHODOLOGIE STATISTIQUE

Ce chapitre est consacré à l'introduction d'une méthode statistique susceptible d'établir une certaine hiérarchie des variables d'entrée. Nous présentons une méthodologie basée sur l'ANOVA généralisée au cas de plusieurs facteurs qui donnera une mesure de sensibilité à chacune des variables d'entrée relativement à la sortie. Nous nous servons également de l'ANOVA factorielle afin de détecter les interactions, pour le calcul des indices d'ordre supérieur. Finalement, l'estimation des quantités D, D_i et D_{ij} définies au chapitre précédent permettra d'approximer les indices de sensibilité totaux.

4.1 Choix de la méthode statistique

Au chapitre précédent, nous avons observé un certain nombre de méthodes d'AS pour la classification de variables. Le choix de la méthode adéquate repose essentiellement sur les spécifications du projet rencontrés en introduction. (De Rocquigny *et al.*, 2008) ont abordé les critères d'un bon choix de méthode. Ceux-ci peuvent se résumer ainsi :

– Les propriétés du modèle : linéaire ou non linéaire, monotone ou non monotone (ce qui pourrait être inconnu *a priori*), le temps CPU ;
– Les caractéristiques des variables d'entrée du modèle : nombre, indépendance ou corrélation, présence de variables discrètes ;
– L'objectif de l'étude et les particulatités de la méthode potentielle : qualitative ou quantitative, locale ou globale, etc.

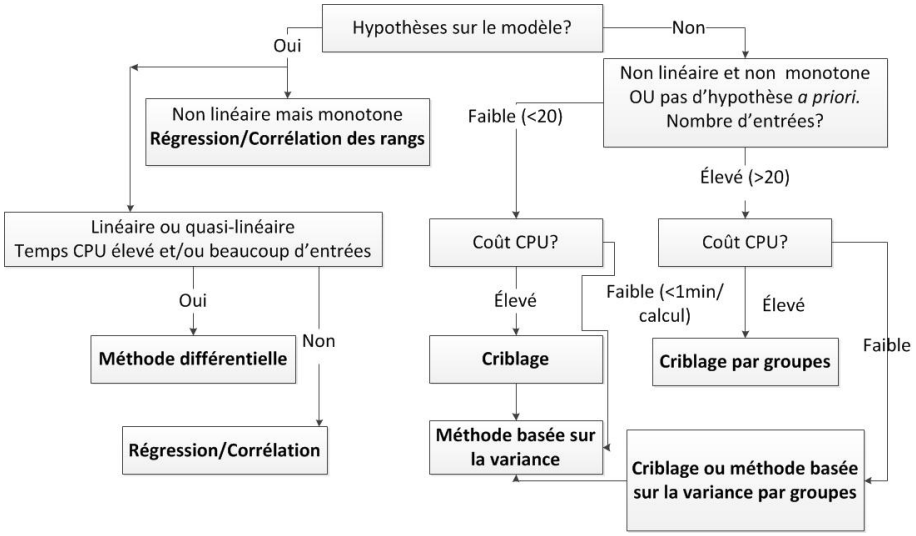

Figure 4.1 Arbre de décision pour le choix de la méthode d'AS appropriée, figure tirée de (De Rocquigny *et al.*, 2008)

La figure 4.1 illustre les critères mentionnés et permet de nous orienter vers les méthodes de criblage ou les méthodes basées sur l'ANOVA par groupes (puisqu'on ne peut émettre d'hypothèse sur le modèle et que le coût CPU de la méthode que nous utiliserons doit être faible). Cependant, compte tenu des spécifications du projet, ceci n'est pas possible soit parce qu'on ne peut pas vérifier certaines hypothèses, par exemple les entrées indépendantes (FAST et Sobol), soit parce que la taille de l'échantillon requise est considérablement grande (d'après (Jacques, 2005), un échantillon de taille 10000 est suffisant pour une bonne estimation des indices de sensibilité de Sobol), ou encore parce que la technique de calcul ne peut être appliquée dans le cadre de l'optimisation des boîtes noires : on parle surtout des échantillons générés en fixant les valeurs d'une entrée tout en variant les autres. Rappelons qu'un échantillon correspond dans notre cas à l'ensemble des solutions visitées par l'algorithme MADS ainsi que les valeurs de retour correspondantes (les sorties de la boîte noire) recensés dans un fichier historique, si on lance celui-ci sur un nombre d'évalutions donné (qui doit être le plus petit possible, car le coût des évaluations est important). C'est à partir de cet échantillon déterministe que

nous sommes censés effectuer une AS.
Compte tenu de ce qui précède, il apparaît que les méthodes d'AS rencontrées auparavant ne sont pas directement applicables dans notre contexte.

4.2 Calcul des indices de sensibilité d'ordre un

Le calcul des indices de sensibilité des variables d'entrée permet d'étudier l'impact de celles-ci sur $Y = f(X_1, X_2, ..., X_k)$. Dans notre cas, Y est la sortie d'une boîte noire, le résultat d'un code informatique. Rappelons la définition de l'indice de sensibilité S_i de la variable X_i quantifiant l'effet de cette dernière sur Y (formule (3.5)) :

$$S_i = \frac{V(E(Y|X_i))}{V(Y)} = \frac{D_i}{D}.$$

Nous présenterons, un peu plus loin dans cette section, une façon d'estimer cette quantité en ayant recours à l'ANOVA.

4.2.1 Analyse de la variance à un facteur

Le principal objectif de l'ANOVA à un facteur X (*one-way* ANOVA) sur une variable Y est d'évaluer l'effet du facteur en comparant les moyennes de Y obtenues selon les modalités de X. La comparaison se fait en examinant la variance des dites moyennes. Cette variance que l'on qualifie de « variabilité inter traitements », constitue une estimation plausible du numérateur D_i de (3.4). On peut donc envisager d'estimer les indices de sensibilité en utilisant la décomposition de la variance donnée par l'égalité fondamentale des modèles de l'ANOVA. Afin d'illustrer la méthode, considérons dans un premier temps le modèle de l'ANOVA à un facteur. Les données à exploiter proviennent d'une étude expérimentale généralement planifiée (exemple : un plan d'expérience). Celles-ci

peuvent être regroupées comme suit :

Tableau 4.1 Données relatives à l'ANOVA à un facteur

Facteur X						
modalité 1	..	..	modalité i	..	..	modalité p
y_{11}			y_{i1}			y_{p1}
.						.
.						.
y_{1j}			y_{ij}			y_{pj}
.						.
.						.
y_{1n_1}			y_{in_i}			y_{pn_p}
$\bar{y}_1$	..	..	$\bar{y}_i$	..	..	$\bar{y}_p$

Le facteur X est le seul facteur contrôlé. Il présente p modalités telles que la $i^{\text{ème}}$, $i = 1, \ldots, p$, comporte n_i répétitions. Ce nombre de répétitions peut donc varier d'une modalité à une autre. Les valeurs de la sortie correspondantes sont présentées dans le tableau 4.1, où y_{ij} est la $j^{\text{ème}}$ valeur de Y obtenue selon la modalité i et $\bar{y}_i$ est la moyenne des valeurs de Y selon la modalité i.
D'après le tableau 4.1, on considère :

$$\bar{y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij}, \text{ et } \bar{y} = \frac{1}{N}\sum_{i=1}^{p}\sum_{j=1}^{n_i} y_{ij}, \text{ avec } N = \sum_{i=1}^{p} n_i.$$

Équation fondamentale de l'ANOVA :

L'ANOVA permet d'expliquer la variance totale d'un échantillon en fonction de la variance due aux facteurs et celle due à l'interaction. L'équation fondamentale de l'ANOVA

(EFAV) décompose la variation totale en plusieurs sources de variation.
Dans le cas du modèle à un seul facteur, l'EFAV est :

$$\underbrace{\sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y})^2}_{SS_{totale}} = \underbrace{\sum_{i=1}^{p} n_i(\bar{y_i}-\bar{y})^2}_{SS_{inter}} + \underbrace{\sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y_i})^2}_{SS_{intra}} \tag{4.1}$$

où :

SS_{totale} est la somme des carrés des écarts totaux ou variation totale. En la divisant par N, on obtient une estimation de la variance $V(Y)$ de Y,

SS_{inter} est la somme des carrés des écarts liés aux effets du facteur A ou variation inter-modalités. Cette somme, divisée par N, constitue une estimation de la variance conditionnelle $V(E(Y|X))$, X étant le facteur (ou la variable) présentée au tableau 4.1,

SS_{intra} est la somme des carrés des écarts résiduels ou variation intra-modalités.

Preuve :

On a

$$y_{ij} = \bar{y} + (\bar{y_i} - \bar{y}) + (y_{ij} - \bar{y_i}),$$

et

$$(y_{ij} - \bar{y}) = (\bar{y_i} - \bar{y}) + (y_{ij} - \bar{y_i}).$$

En élevant les écarts au carré :

$$(y_{ij} - \bar{y})^2 = (\bar{y_i} - \bar{y})^2 + (y_{ij} - \bar{y_i})^2 + 2(\bar{y_i} - \bar{y})(y_{ij} - \bar{y_i}).$$

En faisant la somme sur tous les j :

$$\sum_{j=1}^{n_i}(y_{ij}-\bar{y})^2 = n_i(\bar{y_i}-\bar{y})^2 + \sum_{j=1}^{n_i}(y_{ij}-\bar{y_i})^2 + 2(\bar{y_i}-\bar{y})\sum_{j=1}^{n_i}(y_{ij}-\bar{y_i}).$$

Or $2(\bar{y}_i - \bar{y})\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i) = 0$, car $\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i) = 0$.
En faisant la somme des p modalités, on obtient l'égalité (4.1). □

L'EFAV est vraie dès qu'on dispose de données sous la forme du tableau 4.1, indépendamment des conditions usuelles exigées dans un modèle d'ANOVA (normalité, etc). Des indicateurs d'importance pourront être issus de cette égalité, comme on peut le voir à la suite de cette section.

Rapport de corrélation

Le rapport de corrélation η^2 est une mesure du lien entre deux variables X et Y. X peut être qualitative ou quantitative.
Pour un facteur X, selon (McKay, 1997), l'expression théorique de η^2 est sous la forme :

$$\eta^2 = \frac{V(E(Y|X))}{V(Y)}.$$

Ce qui correspond à l'indice de sensibilité d'ordre un (équation (3.4)) pour le facteur X. (McKay, 1997) a donné une estimation empirique de η^2 basée sur l'ANOVA, en fonction des composantes de l'EFAV.

$$\hat{\eta}^2 = \frac{SS_{inter}}{SS_{totale}}.$$

Compte tenu de l'EFAV, $\hat{\eta}^2$ varie entre 0 et 1. Plus il est proche de 1, plus l'influence de X sur Y est importante.
Nous obtenons ainsi une mesure d'importance qui nous permettra de classer les variables en fonction de leur impact. Cependant, comme nous traitons des problèmes à plusieurs entrées, une généralisation est nécessaire.

4.2.2 Généralisation de l'ANOVA à un facteur

On se place dans le cas général de plusieurs facteurs, c'est à dire $Y = f(X_1, X_2, ..., X_k)$. Aucune hypothèse sur les entrées n'est soumise (indépendance, loi de probabilité, etc).

Alors, on définit l'indice de sensibilité S_i de la variable X_i, $i \in \{1, .., k\}$ par :

$$S_i = \frac{V(E(Y|X_i))}{V(Y)} = \eta_i^2. \tag{4.2}$$

Le tableau 4.1 est simplement étendu au cas de k facteurs à différentes modalités dont on manipule uniquement les valeurs de retour. En effet, on peut remarquer que dans le cas des boîtes noires, pour chaque facteur X_i, l'ensemble des données peut être mis sous la forme du tableau 4.1. L'ensemble des indices du premier ordre définis par (4.2) peut alors être estimé par :

$$\hat{S}_i = \hat{\eta}_i^2, \qquad\qquad i = 1, \ldots, k.$$

Pour les indices d'ordre supérieur, on considère les interactions d'un modèle d'ANOVA factoriel.

4.3 Calcul des indices de sensibilité d'ordre deux

Rappelons que l'indice S_{ij} d'ordre deux représente la contribution due à l'interaction entre les variables X_i et X_j (formule (3.7)) :

$$S_{ij} = \frac{V(E(Y|X_i, X_j)) - V(E(Y|X_i)) - V(E(Y|X_j))}{V(Y)} = \frac{D_{ij}}{D}.$$

À la section suivante, on se propose d'estimer ces indices en se basant sur l'ANOVA factorielle.

4.3.1 Analyse de variance factorielle

On parle de l'ANOVA factorielle lorsque deux facteurs ou plus sont impliqués et que toutes les modalités de tous les facteurs sont utilisés (c-à-d les mesures de Y sont obtenues pour toutes les combinaisons des modalités des facteurs). Le but est d'étudier l'effet des interactions, en plus des effets principaux. À titre d'illustration, nous prsentons le principe du calcul des interactions et des effets principaux dans le cas simple d'un modèle d'ANOVA à deux facteurs A et B, avec le même nombre d'observations pour chaque combinaison de modalités. Le facteur A apparaît sous p modalités et B apparaît sous q modalités, comme le montre le tableau ci-dessous.

Tableau 4.2 Arrangement des données pour un design factoriel à deux facteurs

Facteur A \ Facteur B	1	2	...	q
1	y_{111}, y_{112} ...,y_{11n}	y_{121}, y_{122} ..., y_{12n}	...	y_{1q1}, y_{1q2} ..., y_{1qn}
2	y_{211}, y_{212} ...,y_{21n}	y_{221}, y_{222} ..., y_{22n}	...	y_{2q1}, y_{2q2} ..., y_{2qn}
. . .	. . .	. . .		. . .
p	y_{p11}, y_{p12} ...,y_{p1n}	y_{p21}, y_{p22} ..., y_{p2n}	...	y_{pq1}, y_{pq2} ..., y_{pqn}

Il est démontré que l'EFAV dans le cas de deux facteurs A et B (Montgomery, 2001) est :

$$SS_{totale} = SS_A + SS_B + SS_{AB} + SS_E. \tag{4.3}$$

où :

SS_A est la somme des carrés dus au facteur A. Cette somme équivaut à SS_{inter} de (4.1), avec A comme seul facteur,

SS_B est la somme des carrés dus au facteur B. Cette somme équivaut à SS_{inter} de (4.1), avec B comme seul facteur,

SS_{AB} est la somme des carrés des écarts liés à l'interaction entre A et B,

SS_E est la somme des carrés résiduelle. Cette somme est similaire à SS_{intra} de (4.1).

SS_{AB} et SS_{totale} sont données respectivement par :

$$SS_{AB} = n\sum_{i=1}^{p}\sum_{j=1}^{q}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2,$$

$$SS_{totale} = \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{n}(y_{ijk} - \bar{y}_{...})^2,$$

où :

$$y_{...} = \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{n} y_{ijk},\ \bar{y}_{...} = \frac{y_{...}}{n},$$
$$y_{i..} = \sum_{j=1}^{q}\sum_{k=1}^{n} y_{ijk},\ \bar{y}_{i..} = \frac{y_{i..}}{qn},$$
$$y_{.j.} = \sum_{i=1}^{p}\sum_{k=1}^{n} y_{ijk},\ \bar{y}_{.j.} = \frac{y_{.j.}}{pn},$$
$$y_{ij.} = \sum_{k=1}^{n} y_{ijk} \text{ et } \bar{y}_{ij.} = \frac{y_{ij.}}{n}.$$

On peut donc envisager d'estimer l'indice de sensibilité d'ordre 2, S_{ij} par :

$$\hat{S}_{ij} = \frac{SS_{AB}}{SS_{totale}}.$$

4.3.2 Généralisation de l'ANOVA factorielle

Dans le contexte des boîtes noires où l'expérience n'est pas planifiée, n n'est pas nécessairement partout le même. On définit alors n_{lm}, pour $l \in \{1, .., p\}$ et $m \in \{1, .., q\}$, comme le nombre de valeurs Y obtenues pour la cellule correspondant à la modalité l du facteur A et la modalité m du facteur B. Dans ce contexte, les modalités des facteurs sont les différentes valeurs observées de ceux-ci et les valeurs de Y dans la cellule sont celles observées lorsque le premier facteur (A) vaut l et le deuxième (B) vaut m. Ce nombre de valeurs, noté n_{lm}, n'est donc pas identique pour toutes les combinaisons de modalités des facteurs.

On peut montrer, dans ce cas, que la somme des carrés due à l'interaction (Montgomery, 2001) est :

$$S_{AB} = \sum_{i=1}^{p}\sum_{j=1}^{q} n_{ij}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2.$$

L'indice de sensibilité S_{AB} peut donc être estimé dans ce contexte par :

$$\hat{S}_{AB} = \frac{SS_{AB}}{SS_{totale}} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q} n_{ij}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2}{\sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{n_{ij}}(y_{ijk} - \bar{y}_{...})^2}. \tag{4.4}$$

La formule (4.4) et le fait que $\hat{S}_{AB} = \hat{S}_{BA}$ seront employés afin d'approximer les indices de sensibilité d'ordre 2 définis plus haut ainsi que les indices totaux de la section suivante.

Remarque : Il est possible d'écrire l'équation (4.1) pour un nombre quelconque de facteurs et d'obtenir ainsi les sommes de carrés des interactions de tout ordre. Ce qui permet d'estimer les indices de sensibilité de différents ordres et les indices totaux.

4.4 Indices de sensibilité totaux

Par définition, l'indice de sensibilité total d'une variable X_i est la somme de son indice de premier ordre et de tous les indices d'ordre supérieur où figure cette variable. C'est une autre façon de calculer cet indice dont la logique diffère de l'équation (3.4) du chapitre précédent, où on procède par soustraction.

Par exemple, si le problème est de dimension $k = 3$, l'indice total de la variable X_1 est :

$$S_{T_1} = S_1 + S_{12} + S_{13} + S_{123}.$$

Étant donné le temps de calcul important au delà des indices d'ordre deux, on approxime les indices totaux en faisant la somme de l'indice d'ordre un de la variable X_i en utilisant l'équation (4.2) et tous les indices d'ordre deux contenant l'indice i en utilisant l'équation (4.4).
L'indice total pour cet exemple se réduit à :

$$S_{T_1} = S_1 + S_{12} + S_{13}.$$

Exemple :

On considère le problème-test BROWNAL (Gould *et al.*, 2003), pour $k = 3$ variables et 5 évaluations de la sortie, alors on obtient de fichier historique suivant :

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0.5	0.5	0.5	8.765625
1.5	0.5	-1.5	17.515625
1.5	0.5	0.5	1.360625
3.5	0.5	0.5	17.015625
4.5	0.5	0.5	40.015625

Le calcul des indices d'ordre un et d'ordre supérieur donne les valeurs suivantes :

Indices de premier ordre :

$\hat{S}_1 = 0.8454909083$,

$\hat{S}_2 = 0$,

$\hat{S}_3 = 0.0004911682562$.

Le calcul des indices d'ordre un montre que la variable X_1 est prépondérante.

Indices de deuxième ordre :

$\hat{S}_{12} = 0$,

$\hat{S}_{13} = 0.141226193$,

$\hat{S}_{23} = 0$.

Le calcul des indices d'ordre deux montre que les interactions entre les variables X_1, X_2 et X_2, X_3 est nulle.

Indices totaux :

$\hat{S}_{T1} = 0.986717101$,

$\hat{S}_{T2} = 0$,

$\hat{S}_{T3} = 0.141717361$.

Le classement des variables selon les indices totaux approximés est le même que celui d'ordre un. On peut donc prévoir des résultats similaires de ces deux méthodes.

4.5 Application

On considère maintenant un exemple académique. Celui-ci est extrait du guide d'utilisateur de NOMAD (Le Digabel, 2009) afin de calculer les indices de sensibilité d'ordre un selon la formule (4.2). La taille de l'échantillon est 100 (nombre d'évaluations) et $k = 5$.

$$\min_{x \in \mathbb{R}^5} f(x) = x_5$$

sujet à

$$\begin{cases} c_1(x) = \sum_{i=1}^{5} (x_i - 1)^2 - 25 \leq 0 \\ c_2(x) = 25 - \sum_{i=1}^{5} (x_i + 1)^2 \leq 0 \\ x_i \geq -6 \quad i = 1, 2, ..., 5 \\ x_1 \leq 5 \\ x_2 \leq 6 \\ x_3 \leq 7. \end{cases}$$

Il est trivial que x_5 est la variable la plus importante, de même pour tout x_i, tel que $f(x) = x_i$. Le tableau 4.5 montre que l'indice relatif à cette variable est égal à 1. Le classement des autres variables ne peut pas être jugé pour l'instant. Nous ne calculons pas les indices de sensibilité d'ordre deux pour cet exemple, puisque nous ne pouvons pas vérifier l'exactitude de nos résultats.

Tableau 4.3 Indices de sensibilité d'ordre un pour $f(x) = x_i$,$i \in \{1, 2, 3, 4, 5\}$

$f(x)$		x_1	x_2	x_3	x_4	x_5
Formule (4.2)	S_1	1	0.067	0.234	0.202	0.186
	S_2	0.1726	1	0.166	0.214	0.223
	S_3	0.2529	0.209	1	0.133	0.2104
	S_4	0.2312	0.1126	0.2540	1	0.2880
	S_5	0.2918	0.1805	0.2830	0.3625	1

Les mêmes formules seront utilisées aux chapitres 5 et 6 pour le classement des variables en fonction de leur influence sur la sortie. Plusieurs stratégies seront présentées au cours du chapitre 5 et testées au chapitre 6, où le classement des variables se fait en fonction des indices d'ordre un ou des indices totaux.

CHAPITRE 5

IMPLÉMENTATION CONCEPTUELLE DE STATS-MADS

STATS-MADS est le nom que nous avons donné à l'algorithme qui applique la méthode statistique vue au chapitre précédent sur les données de l'algorithme MADS (dont on évalue les sorties) évoqué au chapitre 2, permettant ainsi l'optimisation alternée entre des sous-espaces de variables et l'espace complet. Dans ce chapitre, nous présentons une manière de concevoir cet algorithme. Dans un premier lieu, nous introduisons une instantiation STATS-MADS basique dont la structure générale est construite en fonction des problèmes tests. Celui-ci ne fait intervenir que les indices de sensibilité d'ordre un pour le classement de variables. Dans un deuxième lieu, nous tentons d'incorporer les indices d'ordre deux pour en évaluer la portée sur la précision du classement de variables. Nous présentons également différentes variantes du STATS-MADS basique.

Note : Afin de ne pas confondre le compteur d'itérations k avec le nombre de variables du problème d'optimisation, ce dernier sera noté n.

5.1 Le logiciel NOMAD

NOMAD (pour *Nonlinear Optimization with the* MADS *Algorithm*) ((Le Digabel, 2011), (Le Digabel, 2009), (Abramson *et al.*, 2012)) est une implémentation C++ de l'ensemble des algorithmes MADS (Audet et Dennis, Jr., 2006). Il est conçu idéalement pour la résolution de problèmes d'optimisation sous contraintes de boîtes noires sous la forme suivante rencontrée auparavant :

$$\min_{x \in \Omega} f(x)$$

où : $\Omega = \{x \in X : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n, f, c_j : X \to \mathbb{R} \cup \{\infty\}$, pour tout $j \in J = \{1, 2, \ldots, m\}$ et X est un sous-ensemble de $\mathbb{R}^n$.

GPS ((Torczon, 1997), (Audet et Dennis, Jr., 2003)) fut la première implémentation dans NOMAD. Il en constitue maintenant une option. ((Le Digabel, 2011), (Le Digabel, 2009)) décrivent toutes les fonctionnalités de NOMAD, les extensions de MADS ainsi que la procédure d'installation et les différents modes. Il existe aussi deux implémentations MATLAB de MADS, l'une est appelée NOMADm (Abramson, 2004), l'autre est disponible dans les outils GADS de MATLAB (MathWorks, 2005).

Paramètres par défaut de NOMAD

Nous nous intéressons aux paramètres les plus importants de NOMAD dans les conditions standards. Comme nous comptons traiter nos problèmes avec la barrière progressive PB ((Audet et Dennis, Jr., 2009)), nous exposons les paramètres qui lui sont reliés.

– L'instance ORTHOMADS : tel que mentionné au chapitre 2, ORTHOMADS (Abramson *et al.*, 2009) est l'algorithme par défaut de NOMAD. ORTHOMADS génère au plus $2n+2$ directions d'exploration à chaque itération, utilisant une approche opportuniste selon laquelle la sonde s'achève dès qu'un point d'essai améliorant est trouvé.
– PB : trois paramètres intercèdent dans le cas de PB :
  • H_MAX_0 : c'est la valeur initiale de $h\_\max$ telle que si pour un point d'essai t $h(t) > h\_\max$, alors t est rejeté. Sa valeur est fixée à 1E+20.
  • H_MIN : c'est la valeur de $h\_\min$ pour laquelle t est considéré réalisable si $h(t) \leq h\_\min = 0$.
  • H_NORM : elle permet de calculer $h(x)$ et correspond à la norme 2.
– Le paramètre de taille initiale du treillis Δ_0^m : c'est un vecteur de n éléments. Pour des raisons de mise à l'échelle, les différentes variables n'ont pas nécessairement la même taille. Si cette quantité n'est pas introduite par l'utilisateur, alors la valeur par défaut

est utilisée et est relative aux bornes selon 5.1.

$$\Delta_0^m = \begin{cases} \frac{(Ub-Lb)}{10} & \text{si } Lb \leq x \leq Ub, x \in X \\ \max\{|x_0|, 1\}, & \text{sinon.} \end{cases} \tag{5.1}$$

– La recherche spéculative : dans le but de promouvoir une direction de succès d d'une itération antérieure $k-1$, MADS évalue un point t_k du treillis ($t_k = x_{k-1} + 4\Delta_k^m d$) au cours de l'étape de recherche. Si $\Delta_k^m < 1$, alors le point évalué est : $t_k = x_{k-1} + \Delta_k^m d$. Cette condition garantit l'appartenance de t_k au treillis courant d'après (Audet et Dennis, Jr., 2006).

5.2 Principe de STATS-MADS

L'idée de base de STATS-MADS est d'amorcer des processus d'optimisation dans des sous-espaces différents, à partir de l'espace plénier de variables. Cette technique est décrite de façon très générale par l'algorithme 3, où J_k représente le sous-ensemble d'indices de variables libres à l'itération k. Cependant, la méthode adoptée au chapitre 4 permettant le classement des variables exige qu'on alterne entre l'espace de n variables et celui de dimension $|J_k|$. En effet, si à l'itération k on fixe $|\bar{J}_k|$ variables (J_k et $\bar{J}_k$ forment une partition de $I = \{1, 2, .., n\}$), alors les indices de sensibilité relatifs à ceux-ci sont nuls, étant donné que $V(E(Y|X_s)) = 0$, pour tout $s \in \bar{J}_k$.

Par conséquent, selon la logique de l'algorithme 3, on fixera toujours les mêmes variables (d'indice de sensibilité nul), ce qui correspond à une des variantes de STATS-MADS dont on discutera plus loin.

Algorithme 3 : Principe global de STATS-MADS

1 : Initialisation : $I = \{1, 2, .., n\}$, n variables d'entrée, $k \leftarrow 0$;
2 : Lancer MADS sur les variables de I ;
3 : Trouver $J_k \subset I$ par une analyse statistique ;
4 : Lancer MADS sur J_k ;
5 : Retourner à 3 avec $k \leftarrow k + 1$.

5.3 Les ingrédients de STATS-MADS

Nous allons maintenant détailler les composants de STATS-MADS. Nous considérons les notations supplémentaires ci-dessous qui y sont spécifiques.

Terminologie spécifique à STATS-MADS

Par souci de simplification, les appellations MADS et ORTHOMADS seront désormais confondues. Afin d'alléger l'écriture, un MADS-RUN désigne un appel à NOMAD pour l'exécution de l'algorithme MADS. On considère les paramètres par défaut de NOMAD (version de développement $3.5.2$), à l'exception des modèles quadratiques (étape de recherche) qui sont rendus inactifs (notons qu'ils le sont déjà pour les problèmes de dimension supérieure à 50) et de la génération de points à l'extérieur des bornes inférieures et supérieures (paramètre : SNAP_TO_BOUNDS) qui devient permise. Ces exceptions sont justifiées par une meilleure optimisation perçue à la phase des tests qui affecte la gestion de la taille du cadre et du treillis.

Un MADS-RESCUE désigne un recours à MADS-RUN, dans le cas où la stratégie de fixation de variables aboutit à un état de stagnation. Par stagnation, on entend que le programme est coïncé à un minimum local. Dans MADS-RESCUE, toutes les variables sont relâchées et MADS-RUN est lancé avec un nombre d'évaluations égal à un entier M, founi par l'utilisateur. Ce nombre doit être suffisamment grand afin de permettre, le cas échéant, de franchir un minimum local. À chaque recours à MADS-RESCUE, le

nombre d'évaluations est multiplié par une constante τ ($M \leftarrow \tau M$, $\tau > 1$). En effet, chaque recours signifie que, s'embarquer dans un sous espace, n'est pas un processus optimisant. Il est plus intéressant, dans ce cas-ci, d'augmenter le nombre d'évaluations dans l'espace de l'ensemble des variables. La figure 5.1 illustre la descente locale du problème BROWNAL (Gould *et al.*, 2003), pour un nombre de variables égal à 20 et un pourcentage de fixation égal à 80%. La figure montre un comportement similaire sur les 774 premières évaluations. Pour les évaluations suivantes, le MADS-RESCUE fuit l'état de stagnation en lançant un grand nombre d'évaluations (M) sur l'espace complet, d'où l'intérêt de cette approche de secours.

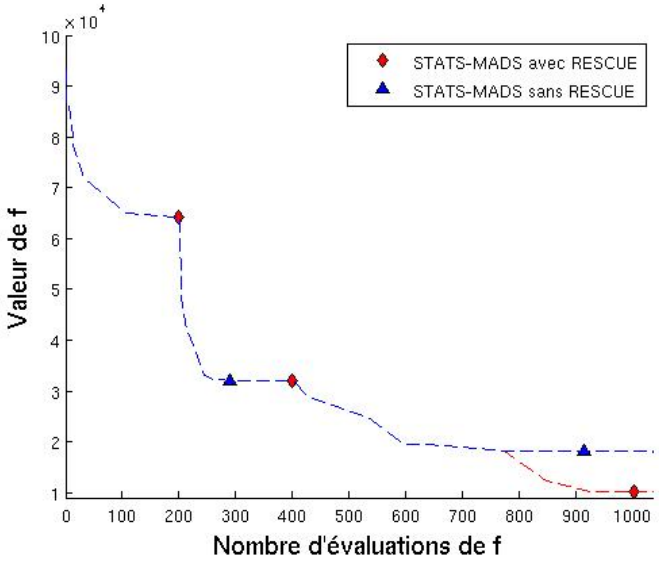

Figure 5.1 Utilité du processus MADS-RESCUE

On définit alors trois processus pour STATS-MADS :

- P_1 : MADS-RUN avec variables relâchées : espace complet.
- P_2 : MADS-RUN avec variables fixes : sous-espace.
- P_3 : MADS-RESCUE : espace complet.

Lors de l'exécution de STATS-MADS, on est forcément impliqué dans un des trois processus mentionnés, commençant toujours par P_1 ensuite P_2 sans être deux fois de suite impliqué dans un espace ou dans un sous-espace (voir l'algorithme 4).

Algorithme STATS-MADS **basique**

Rendu ici, on est en mesure de présenter les principales étapes de STATS-MADS. Celui-ci optimise en alternance dans des espaces et des sous-espaces. Le passage d'un espace à un sous-espace se fait au moyen d'une procédure de fixation de variables basée sur le calcul des indices de sensibilité d'ordre un (algorithme 5). Le passage inverse est conditionné par le succès ou l'échec du processus P_2. Dans l'affirmative, on retourne au processus P_1, sinon on fait recours à P_3.

Algorithme 4 : Algorithme STATS-MADS de haut niveau

1 : Effectuer un MADS-RUN sur l'espace complet ;
2 : Fixer un pourcentage de variables les moins influentes (voir algorithme 5) ;
3 : Effectuer un MADS-RUN sur le sous-espace de variables importantes ;
4 : Tester si le processus en sous-espace est un succès. Si oui, aller à 5, sinon aller à 6 ;
5 : Relâcher les variables fixes et refaire un MADS-RUN sur l'espace complet, retourner à l'étape 2 ;
6 : Lancer un MADS-RESCUE et retourner à l'étape 2.

L'algorithme 4 peut être représenté par la figure 5.2 faisant intervenir les critères d'arrêt de chacun des processus. En effet, les étapes 5 et 6 de cet algorithme sont des processus travaillant dans des espaces complets (P_1 et P_3 respectivement). La seule différence figure au niveau des critères de terminaison que nous détaillerons plus loin dans ce chapitre.

Algorithme 5 : Fixation des variables les moins influentes

1 : Charger la cache et récupérer les données ;

2 : Calculer les indices de sensibilité de premier ordre de toutes les variables ;

3 : Trier les indices par ordre croissant ;

4 : Fixer le pourcentage de variables spécifié en considérant la partie entière «plafond».

À l'étape 4 de l'algorithme 5, le nombre de variables à fixer est calculé à partir d'un pourcentage identifié par l'utilisateur et est arrondi à l'entier qui lui est immédiatement supérieur ou égal. Les variables les moins infuentes sont fixées, tout au long de P_2, à la valeur de la dernière solution obtenue et sont relâchées au début du processus suivant.

Notations supplémentaires

Passons maintenant aux détails plus techniques. Définissons alors les paramètres suivants :

- n^p_{eval} : est le nombre d'évaluations « prévu » de f (identifié par l'utilisateur) pour un MADS-RUN (les processus P_1 et P_2) ;
- n^{max}_{eval} : est le nombre maximum d'évaluations de f pour un processus P_i, $i \in \{1, 2, 3\}$. Il vérifie les inégalités suivantes qui seront expliquées à la sous-section Pseudo-code ;

$$n^{max}_{eval} \leq n^p_{eval}, \text{ pour } P_1 \text{ et } P_2. \tag{5.2}$$

$$n^{max}_{eval} \leq M \times \tau^j, j \in \{0, 1, 2 \ldots\}, \text{ pour } P_3. \tag{5.3}$$

  Le compteur j est égal à zéro la première fois qu'on fonce vers P_3 et s'incrémente à chaque retour.
- Δ^m_c : est la taille du treillis courante obtenue au bout d'au plus n^{max}_{eval} évaluations.
- Δ^p_c : est la taille de cadre courante obtenue au bout d'au plus n^{max}_{eval} évaluations.
- n^{tot}_{eval} : est le nombre d'évalutions global qui constitue le budget des évaluations dont

on dispose ;

– n_{eval}^{cumul} est le nombre d'évaluations de tous les processus calculé d'une manière cumulative. n_{eval}^{tot} en est une borne supérieure.
– x_0 : est la solution initiale du problème. Pour des exemples académiques, celle-ci est choisie arbitrairement et est fixée afin de faciliter la comparaison des algorithmes. En industrie, elle est souvent produite par les ingénieurs travaillant sur ces problèmes ;
– $\hat{x}_c$: est la meilleure solution réalisable courante ;
– $\hat{x}_p$: est la meilleure solution réalisable du processus précédent ;
– Le pourcentage de fixation sera simplement appelé $pourcentage$.

Nous utilisons ces notations afin de décrire le fonctionnement des processus à l'algorithme 6, qui représente une implémentation non formelle de STATS-MADS. Certaines conditions gèrent le passage d'un processus à un autre. Elles sont listées au paragraphe suivant.

Critères d'arrêt

Trois critères de terminaison interviennent dans le cas de STATS-MADS :

→ **Critère 1 :** Une limite sur le nombre total d'évaluations de f (locale : n_{eval}^{max} et globale : n_{eval}^{tot}) ;

→ **Critère 2 :** Une tolérance sur la taille du treillis : au bout d'un certain nombre d'évaluations, la taille du treillis devient plus petite que la précision de NOMAD ;

→ **Critère 3 :** Un nombre maximum n_e d'itérations échouées (fixé par l'utilisateur).

Tous les processus peuvent être interrompus par les deux premiers critères. Cependant, seulement les processus P_2 et P_3 le sont pour le troisième, tel que le montre la figure 5.2. Le processus P_2 est qualifié de succès (voir figure 5.2), s'il n'a pas été suspendu par un des critères 2 ou 3, autrement dit, si exactement n_{eval}^{max} évaluations sont accomplies, ou si la meilleure solution réalisable courante $\hat{x}_c$ obtenue est différente de celle du processus

précédent $\hat{x}_p$.

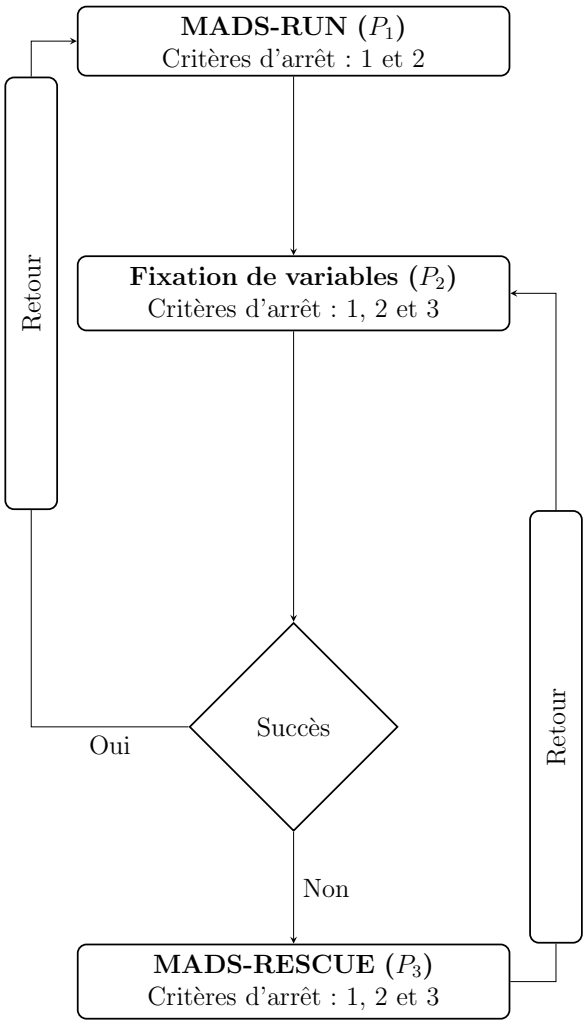

Figure 5.2 Organigramme représentant le fonctionnement général de STATS-MADS

5.3.1 Pseudo-code

L'algorithme 6 suivant décrit le pseudo-code de STATS-MADS.

Algorithme 6 : Pseudo-code du programme principal

Entrées : cache initiale obtenue à partir d'un MADS-RUN sur le problème entré par l'utilisateur, Δ_c^m, Δ_c^p.

Sorties : selon les spécifications de l'utilisateur.

Initialisation : $x_0 \leftarrow \hat{x}_c$;

$\Delta_0^m \leftarrow \Delta_c^m$; $\Delta_0^p \leftarrow \Delta_c^p$;

$n_{eval}^{cumul} \leftarrow taille\_courante\_de\_la\_cache$; $fixation \leftarrow vrai$;

$rescue \leftarrow faux$; $i \leftarrow 0$;

Début d'un processus;

tant que $(n_{eval}^{cumul} < n_{eval}^{tot})$ **faire**

si *rescue=faux* **alors**

$n_{eval}^{max} = \min\left\{n_{eval}^{tot} - n_{eval}^{cumul}, n_{eval}^{p}\right\}$;

si $fixation = vrai$ **alors**

Appeler la fonction qui fixe le pourcentage précisé de variables ;

sinon

Relâcher toutes les variables ;

fin

Fixer le nombre d'échecs consécutifs ;

sinon

$n_{eval}^{max} = \min\left\{n_{eval}^{tot} - n_{eval}^{cumul}, M \times \tau i\right\}$;

Fixer le nombre d'échecs consécutifs;

$i \leftarrow i + 1$;

$rescue \leftarrow faux$;

fin

MADS-RUN ;

si $fixation = faux$ **alors**

$\Delta_0^m \leftarrow \Delta_c^m$;

$\Delta_0^p \leftarrow \Delta_c^p$;

fin

si $((fixation = vrai)$ ***et*** $(taille\_courante\_de\_la\_cache < n_{eval}^{max}$ ***ou*** $\hat{x}_c = \hat{x}_p))$ **alors**

$rescue \leftarrow vrai$;

fin

$fixation \leftarrow !fixation$

Mise à jour du nombre d'évaluations total et du point initial;

$x_0 \leftarrow \hat{x}_c$; $n_{eval}^{cumul} \leftarrow n_{eval}^{cumul} + taille\_courante\_de\_la\_cache$;

fin

Explication du pseudo-code

L'utilisateur doit au préalable définir sa boîte noire avant que l'algorithme 6 ne soit exécuté. On entend par définition de la boîte noire la déclaration de la fonction objectif, des contraintes, du point initial, etc. La procédure détaillée est décrite dans le guide d'utilisateur de NOMAD (Le Digabel, 2009) à sa quatrième section.

Tous les paramètres de la boîte noire sont ressaisis par la suite d'une manière indirecte, étant donné la non accessibilité, au niveau du code source, à celle-ci. Par la suite, un MADS-RUN permettra d'obtenir la première cache à partir de laquelle une analyse de sensibilité sera effectuée (la fonction qui fixe les variables est ainsi appelée).

La variable n_{eval}^{cumul} est mise à jour à chaque fois qu'un processus est exécuté. Pour ce faire, on lui assigne la taille de la cache courante ($taille\_courante\_de\_la\_cache$) à l'étape de l'initialisation ainsi qu'à la fin de chaque processus. Les variables $fixation$ et $rescue$ sont booléennes. La valeur logique « vrai » est affectée à $fixation$ à l'initialisation afin d'activer le processus P_2. Cependant P_3 n'est déclenché que lorsque $rescue$ reçoit « vrai » (c'est l'état de stagnation dont les conditions ont été expliquées plus tôt). La ressaisie de $\hat{x}_c$ est nécessaire à la fin de chacun des processus, car c'est le centre du cadre du processus suivant (le nouveau x_0). Par contre, celle de la taille du cadre et du treillis n'est faite que pour les espaces complets (à la fin de P_1 et P_3), comme suit :

À l'étape 1 de l'algorithme 4, les paramètres Δ_0^m et Δ_0^p sont les valeurs par défaut de NOMAD. Ensuite, aux étapes 3, 5 et 6, les valeurs initiales de ces paramètres sont choisies comme étant égales à leurs valeurs finales obtenues à la conclusion du dernier processus ayant travaillé sur l'espace complet.

Avant de lancer NOMAD avec un processus donné, il est crucial de vérifier si on ne dépasse n_{eval}^{tot} évaluations. Pour cela n_{eval}^{max} est ajusté en fonction de n_{eval}^{cumul} et n_{eval}^{p} pour P_1 et P_2 et n_{eval}^{cumul} et $M \times \tau^j$ pour P_3 (voir algorithme 6). C'est ce qui explique les conditions (5.2) et (5.3).

Ayant défini sa boîte noire, l'utilisateur peut appeler l'algorithme STATS-MADS directe-

ment en spécifiant les paramètres ci-dessous :

STATS-MADS (n^p_{eval}, $pourcentage$, n^{tot}_{eval}, M, τ, n_e).

Note : Pour des fins de simplification, n^p_{eval} est le même pour les processus P_1 et P_2 (P_3 est caractérisé par M et τ).

5.4 Analyse de la convergence de STATS-MADS

Tout comme dans les analyses de convergence de MADS, on examine le comportement lorsque le compteur d'itérations k tend vers l'infini. On suppose qu'il n'y ait pas de critère d'arrêt global (n^{tot}_{eval}) et que STATS-MADS est lancé sur un ensemble fini de sous-espaces de dimensions $|J_q|, q \in \{1, 2, \ldots, p\}$, où $J_q \subset I = \{1, 2, .., n\}$, alors si P_2 est un succès, l'optimisation se fait dans un sous-espace jusqu'à ce qu'elle soit interrompue par le critère 1 ($n^{max}_{eval} = n^p_{eval}$). Dans le cas contraire, elle le sera par les critères 2 ou 3 (n_e). La gestion de la taille du treillis assure que le programme ne s'achève pas dans un sous-espace (où il peut exister une direction au point limite pour laquelle la dérivée de Clarke est négative). L'optimisation dans l'espace de n variables (P_1 et P_3) est alors incontournable. Par conséquent, on est ramené à l'algorithme MADS (Audet et Dennis, Jr., 2006) classique duquel on hérite les principaux résultats de convergence.

5.5 Extensions de STATS-MADS

Nous proposons des variantes du STATS-MADS basique (algorithme 4) dont le but est de diversifier les techniques et d'identifier, plus tard, la meilleure stratégie.

Un STATS-MADS non itératif

STATS-MADS est un algorithme itératif qui passe d'un espace à un sous-espace. Dans le cas d'un STATS-MADS non itératif, la fixation de variables ne se fait qu'une seule fois (un seul P_2). Le processus P_3 est alors absent. L'algorithme passe de P_1 à P_2 et revient à P_1 jusqu'à ce qu'un critère de terminaison soit déclenché.

Un STATS-MADS homogène

Dans le cas d'un STATS-MADS homogène, les sous-espaces de variables sont identiques. Tous les processus sont présents. L'algorithme 5 n'est appelé qu'une seule fois dans l'algorithme 4. On suppose, pour cette stratégie, qu'il existe un seul groupe de variables importantes à travers lequel on lance plusieurs processus P_2 optimisants.

Méthode des indices totaux

Cette méthode consiste à effectuer une modification au calcul des indices de sensibilité. Nous utilisons les indices de sensibilité totaux afin de trier les variables. Ceux-ci sont approximés en faisant la somme de l'indice d'ordre un et tous les indices d'ordre deux relatifs à la variable concernée (chapitre 4). L'algorithme 4 demeure intact . Le seul changement se situe au niveau de l'étape 2 de l'algorithme 5, puisque la formule du calcul des indices englobe dorénavant les indices d'ordre supérieur.

Méthode de la frontière

Il s'agit d'un STATS-MADS basique avec une stratégie de fixation de variables différente. En effet, nous ne nous contentons pas du tri croissant des indices de sensibilité d'ordre un. Nous procédons par une vérification qui concerne les variables qui se situent de part et d'autre de la frontière entre les variables à fixer et les variables qui ne le seront pas. Le nombre de celles-ci n'est pas constant, vu que le pourcentage de fixation est à présent variable.

L'algorithme 5 devient :

Algorithme 7 : Fixation des variables selon la méthode de la frontière

1 : Charger la cache et récupérer les données ;

2 : Calculer les indices de sensibilité de premier ordre de toutes les variables ;

3 : Trier les indices par ordre croissant ;

4 : Déterminer le nombre 2α de variables pour lesquels on calculera les indices totaux : $\alpha = \lceil \frac{\min\{n_1, n_2\}}{4} \rceil$, où n_1 est le nombre de variables à fixer et $n_2 = n - n_1$;

5 : Fixer les $n_1 - \alpha$ variables correspondant aux plus faibles valeurs du tri de l'étape 4 ;

6 : Trier de nouveau les indices calculés à l'étape 4 et en fixer la moitié.

Exemple : Si $n = 10$ et $pourcentage = 30$, alors $n_1 = 3$ et $\alpha = 1$.
À l'étape 5, on fixe $n_1 - \alpha = 2$ variables. À l'étape 6, on trie deux variables et on fixe la plus petite.

Pour clore ce chapitre, notons que nous testons l'ensemble des stratégies présentées au cours du chapitre suivant. Les résultats des tests pourront trancher quant à l'efficacité du STATS-MADS basique ainsi que ses extensions.

CHAPITRE 6

TESTS ET RÉSULTATS NUMÉRIQUES

Ce chapitre vise à présenter l'ensemble des résultats de notre recherche. D'abord, nous appliquons notre algorithme sur un ensemble de problèmes tests issus de la littérature afin de pouvoir tirer des recommandations. Ensuite, nous exposons les résultats des différentes extensions de STATS-MADS rencontrées au chapitre 5. Enfin, nous testons notre méthode sur un problème test réputé difficile, ayant les caractéristiques d'une boîte noire. Nous adoptons toutes les notations du chapitre précédent. L'algorithme STATS-MADS basique sera désigné par STATS-MADS.

6.1 Profils de performance

Afin de comparer les différents algorithmes (MADS, GPS et STATS-MADS), nous nous servons de l'outil présenté dans (Dolan et Moré, 2002). Un profil de performance y est défini comme étant une fonction cumulative d'une performance métrique permettant d'évaluer le record d'un algorithme.

On considère un ensemble de problèmes tests $\mathcal{P}$ de cardinal n_p et un ensemble d'algorithmes $\mathcal{A}$. La performance de l'algorithme $a \in \mathcal{A}$ est évaluée, dans notre cas, en fonction de la valeur de la fonction objectif f_a^* obtenue au bout d'un total de n_{eval}^{tot} évaluations. Cette valeur est considérée comme la valeur optimale produite par l'algorithme a.

Pour $p \in \mathcal{P}$ donné, on pose $f_{p,a}^*$, la meilleure des valeurs de la fonction objectif des algorithmes comparés.

$$f_{p,a}^* = \min_{a \in \mathcal{A}} \ f_a^*.$$

La valeur $f_{p,a}^*$ peut être positive ou négative dépendamment du problème en question. On caractérise le succès ou l'échec de $a \in \mathcal{A}$ par la fonction de score $r_{p,a}$ ci-dessous :

$$r_{p,a} = \begin{cases} 1 & \text{si } f_a^* \leq f_{p,a}^* + \alpha |f_{p,a}^*|, \\ 0 & \text{sinon.} \end{cases}$$

où α est un scalaire positif.

La performance ρ d'un algorithme a est alors définie par :

$$\rho_a(\alpha) = \frac{1}{n_p} \sum_{p \in \mathcal{P}} r_{p,a}.$$

$\rho_a : \mathbb{R}^+ \longmapsto [0, 100]$ est une fonction non décroissante dont la valeur à l'origine représente le pourcentage qu'un algorithme a l'emporte sur tous les autres algorithmes pour l'ensemble des problèmes testés et dont la valeur en $\alpha > 0$ permet de quantifier l'écart relatif à la valeur optimale $f_{p,a}^*$.

6.2 Tests exploratoires

Nous procédons par des tests exploratoires, étant donné que nous ne disposons d'aucune information, *a priori*, sur l'efficacité de l'approche statistique. Aucun plan de tests n'est donc établi.

Dans cette section, nous nous intéressons à un ensemble diversifié de problèmes (lisses ou non, contraints ou non, bornés ou non) dont la taille varie entre 10 et 500. Ces problèmes proviennent de (Lukšan et Vlček, 2000), (Gould *et al.*, 2003), (Audet *et al.*, 2008c) et (Audet et Dennis, Jr., 2009). Leurs caractéristiques sont recensées dans (Conn et Le Digabel, 2011). Tous les points de départ sont donnés. Notons que certains problèmes sont de taille ajustable.

La première gamme de tests comporte un total de 76 problèmes répartis comme suit :

– 13 de dimension $n = 10$;

– 11 de dimension $n = 20$;

– 14 de dimension $n = 50$;

– 14 de dimension $n = 100$;

– 24 de dimension $n = 250$ et $n = 500$.

Sauf indication contraire, tous les tests ont été lancés avec les paramètres suivants : $n^{tot}_{eval} = 100n$, $n^{p}_{eval} = 10n$, $n_e = 3$, $M = 1000$ et $\tau = 4$.

Nous accordons plus de flexibilité au paramètre $pourcentage$ en faisant varier celui-ci entre 10 et 90, avec un écart de 10. Par souci de clarté, nous ne présentons que les pourcentages 10, 50 et 90. De ce fait, il nous sera possible de visualiser le comportement de STATS-MADS losqu'on fixe les valeurs extrêmes des pourcentages ainsi que leur médiane . Les profils de performance obtenus sont donnés par les figures 6.1, 6.2, 6.3, 6.4 et 6.5.

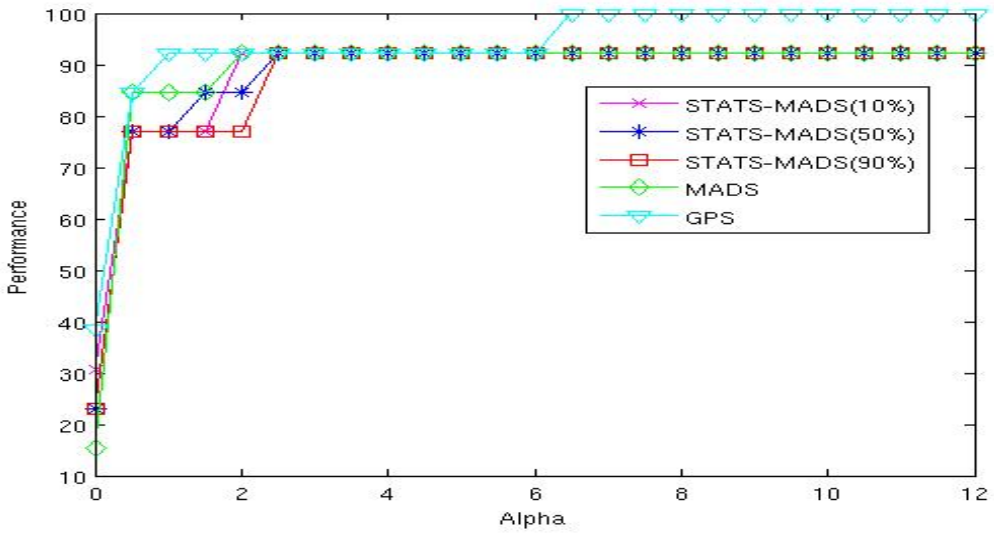

Figure 6.1 Profils de performance dans le cas où $n = 10$ (basés sur 13 problèmes)

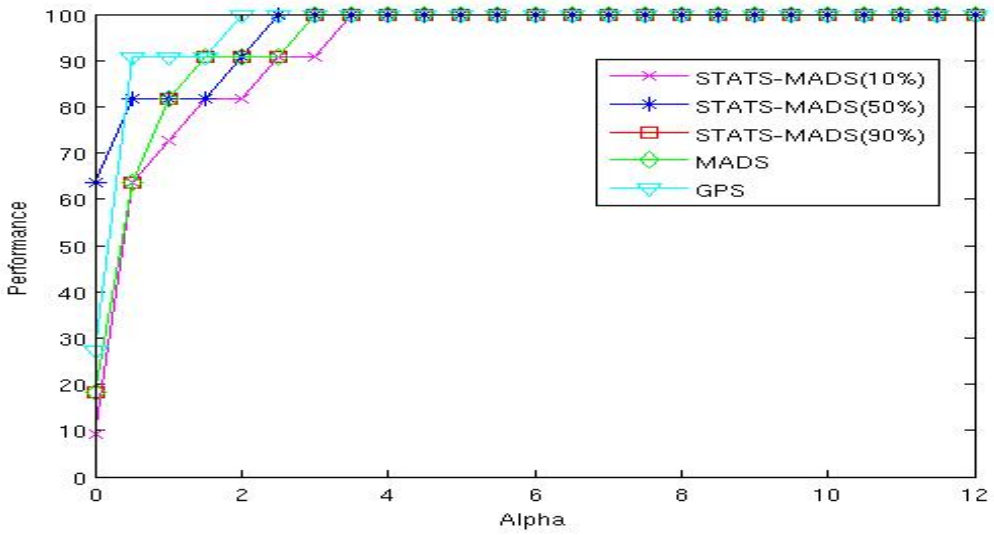

Figure 6.2 Profils de performance dans le cas où $n = 20$ (basés sur 11 problèmes)

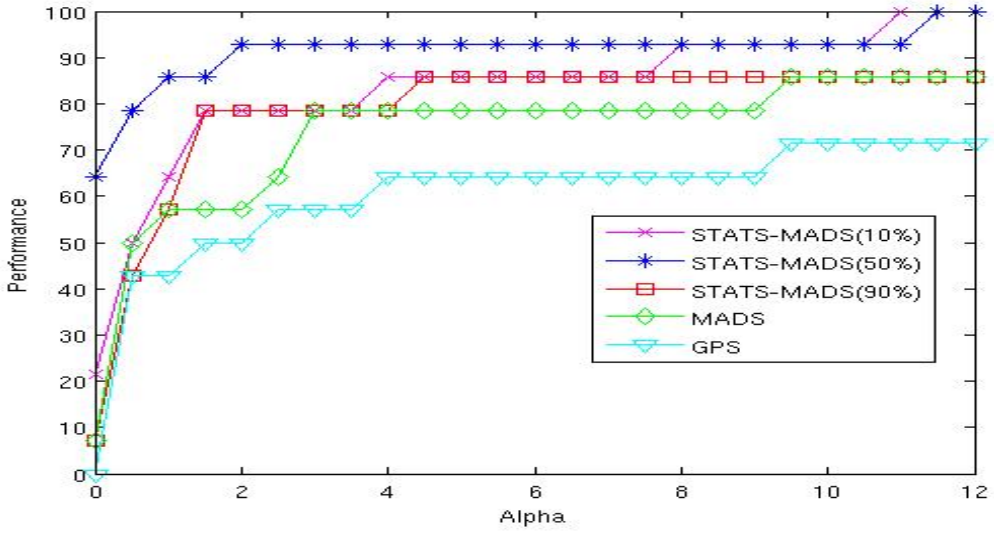

Figure 6.3 Profils de performance dans le cas où $n = 50$ (basés sur 14 problèmes)

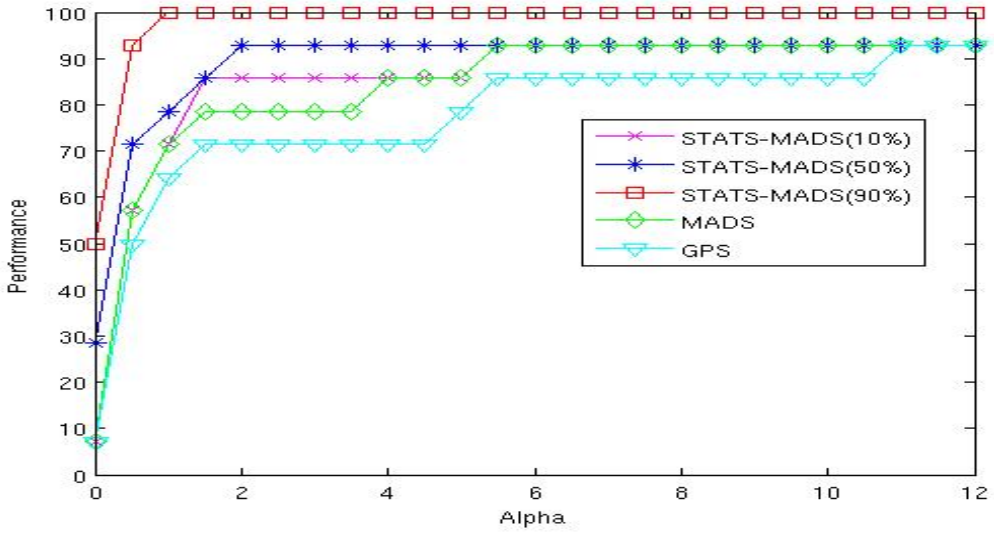

Figure 6.4 Profils de performance dans le cas où $n = 100$ (basés sur 14 problèmes)

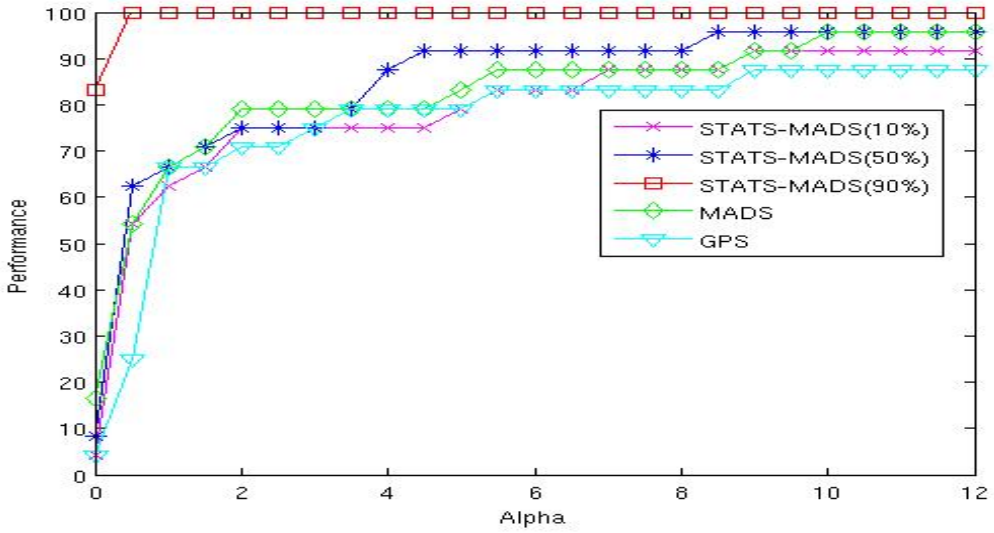

Figure 6.5 Profils de performance dans le cas où $n = 250$ et $n = 500$ (basés sur 24 problèmes)

À la figure 6.1, l'algorithme GPS surpasse légèrements les autres algorithmes. 38.46% des problèmes ont une valeur égale à $f^*_{p,a}$. STATS-MADS(10%) et STATS-MADS(90%)

viennent en deuxième et troisième position respectivement avec 30.77% et 23.08%.
Une différence plus marquée apparaît en augmentant la taille n des problèmes. En effet, à la figure 6.2, c'est STATSMADS (50%) qui détient le meilleur pourcentage (63.64%) suivi par GPS (27.27%) et MADS.
À la figure 6.3, les trois algorithmes STATSMADS (50%), STATSMADS (10%) et STATSMADS (90%) occupent les trois premiers rangs. MADS donne le même résultat que STATSMADS (90%) .
Les figures 6.3 et 6.5 décrivent une meilleure performance de STATSMADS (90%) avec un taux de succès respectivement de 50% et 83.33% et un plafonnement à une valeur $\lambda < 2$.

Afin de diversifier les résultats, nous reprenons l'ensemble des tests avec une stratégie de recherche par hypercube latin (Tang, 1993) non opportuniste avec les paramètres $p_1 = 100n$ et $p_i = 0$, où p_1 est le nombre initial de points d'essai générés à la première itération et p_i est le nombre de points générés pour les itérations $i \geq 2$. Nous obtenons des résultats similaires à ceux de MADS par défaut dont ceux de dimension $n \geq 100$ sont illustrés par les figures 6.6 et 6.7.

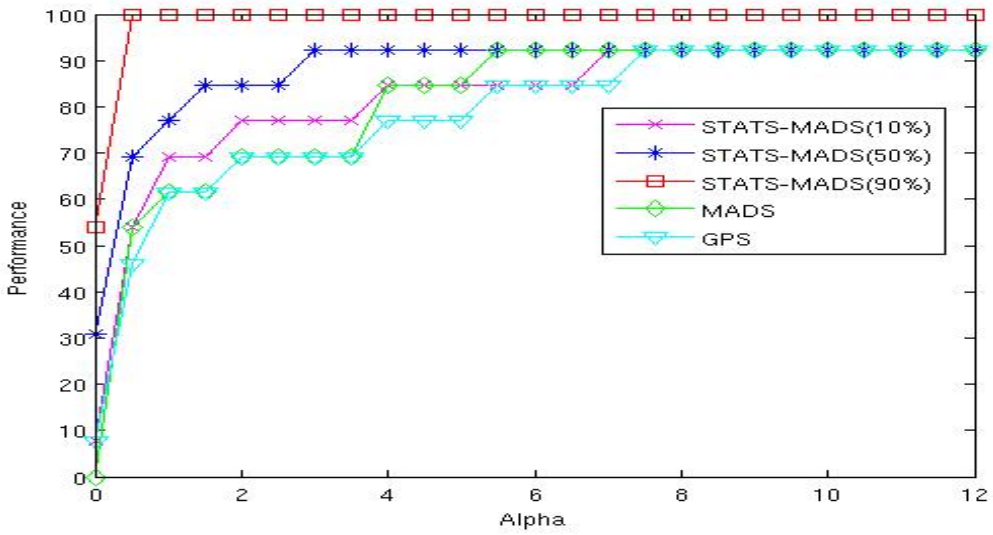

Figure 6.6 Profils de performance dans le cas où $n = 100$ (basés sur 13 problèmes), en utilisant une stratégie de recherche par hypercube latin

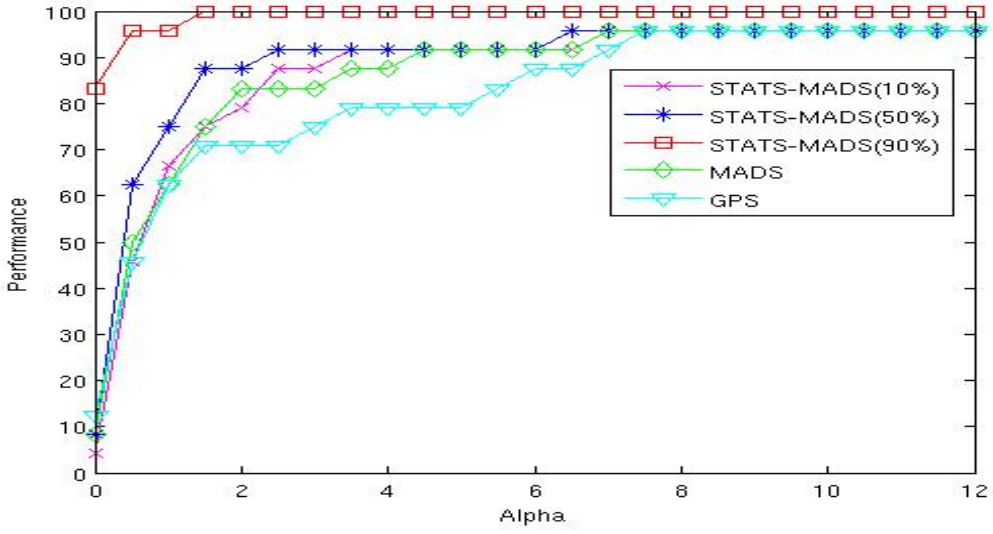

Figure 6.7 Profils de performance dans le cas où $n = 250$ et $n = 500$ (basés sur 24 problèmes), en utilisant une stratégie de recherche par hypercube latin

Premières observations

Les premiers tests montrent une prépondérance des résultats des problèmes de grande dimension, en particulier ceux de dimension $n \geq 250$ avec un pourcentage élevé de variables fixes (90%). La figure 6.8 met en comparaison tous les pourcentages de fixation. Elle justifie bien le choix du pourcentage le plus grand (90%) pour les prochains tests.

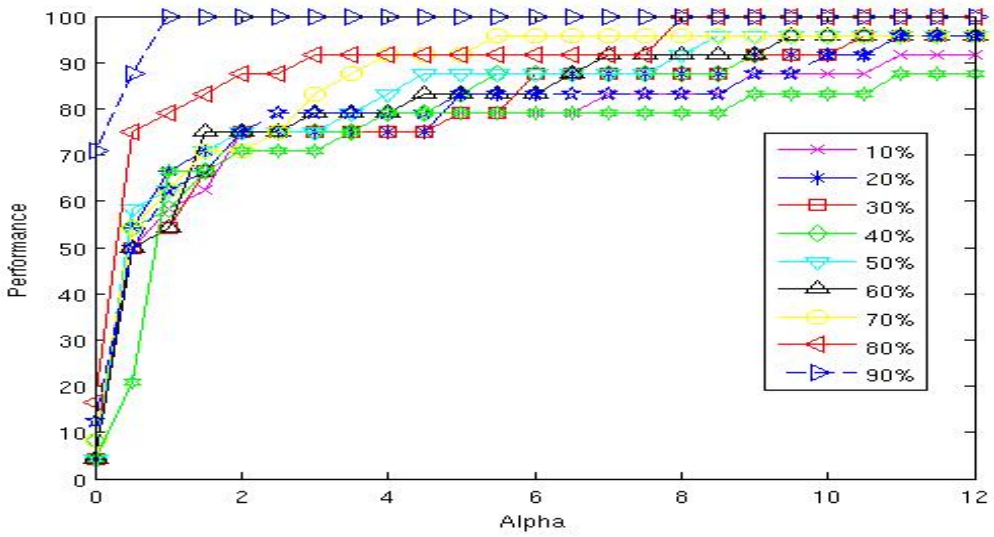

Figure 6.8 Profils de performance illustrants l'importance du choix du pourcentage de fixation pour les problèmes de dimension $n \geq 250$ (24 problèmes)

L'approche paraît donc prometteuse pour les problèmes de grande taille pour lesquels on fixe 90% des variables. Par conséquent, il serait avantageux de focaliser notre intérêt futur sur ce genre de problèmes.

6.3 Cas de problèmes de dimension $n \geq 250$

Nous exécutons l'algorithme STATSMADS sur un ensemble de 12 problèmes tests, en faisant varier le nombre de variables entre 250 et 500 avec un pas de 50 et un pourcentage de fixation de 90, pour chaque problème. Nous obtenons ainsi 72 instances.

Par la suite, nous comparons, pour chaque instance, les valeurs optimales des fonctions objectifs f^* obtenues, au bout de $100n$ évaluations, de MADS, GPS et STATS-MADS, afin que nous puissions évaluer la performance de ce dernier. Les tableaux 6.1 et 6.2 contiennent les valeurs de f^* de ces trois algorithmes. La figure 6.9 illustre une prédominance de STATS-MADS avec un taux de succès supérieur à 80%.

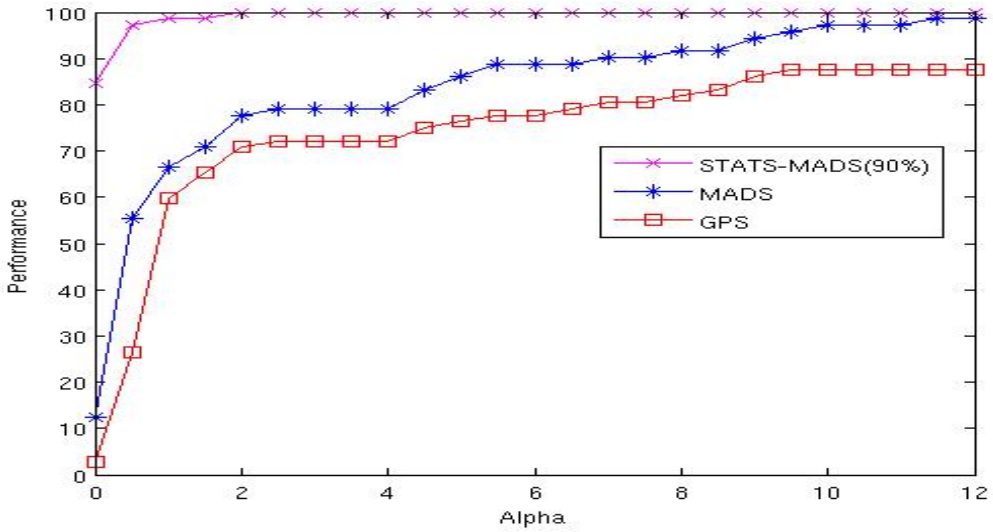

Figure 6.9 Profils de performance dans le cas où $250 \leq n \leq 500$ pour un ensemble de 72 problèmes

Tableau 6.1 Valeurs de f^* des trois algorithmes pour $250 \leq n \leq 500$

Instances		$f^*_{STATS-MADS}$	f^*_{MADS}	f^*_{GPS}
BROWNAL (Gould *et al.*, 2003)	250	-0.19936809634	-0.25493364	-0.0902993563
	300	-0.2307452576	-0.1446767202	-0.03622130354
	350	-0.1103741744	-0.1713176654	-0.08238105208
	400	-0.1162654853	-0.06964743625	-0.03207357373
	450	-0.201875977	-0.1848964899	-0.07618140015
	500	-0.2213701537	-0.2112503316	-0.0734738041
DISK (Audet et Dennis, Jr., 2009)	250	-40.60107422	-25	-15.46875
	300	-41.44921875	-23	-15.46875
	350	-45.23608398	-24.75	-15.46875
	400	-46.59765625	-22.25	-15.46875
	450	-47.71875	-25	-15.46875
	500	-48.3125	-22.5	-15.46875
L1HILB (Lukšan et Vlček, 2000)	250	3.267577877	35.40279223	179.9226579
	300	14.5199942	26.35565427	291.5228721
	350	8.385443251	42.09084886	280.2502033
	400	4.801882716	9.129353313	292.3849055
	450	4.922063162	31.44026852	398.9932892
	500	19.19563522	51.65515457	406.6776583
MXHILB (Lukšan et Vlček, 2000)	250	0.3916285213	0.6677316341	1.049903488
	300	0.02628767687	0.3269626037	1.049542869
	350	0.1609717999	0.1104218969	1.045602413
	400	0.1442806643	0.4665952251	1.045602413
	450	0.05815580861	0.3369307988	1.044279711
	500	0.6171264908	0.7641535173	1.044279711
TRIDIA (Gould *et al.*, 2003)	250	20036.82686	27020	27020
	300	28753.31755	40090	40090
	350	39713.16614	55245	55490
	400	53334.99353	73107.5	73390
	450	68659.40625	93470	93790
	500	86456.10938	116690	116690
G2 (Audet *et al.*, 2008c)	250	-0.1993680963	-0.25493364	-0.0902993563
	300	-0.2307452576	-0.2308798643	-0.08608107767
	350	-0.1103741744	-0.1713176654	-0.08238105208
	400	-0.1162654853	-0.1861061239	-0.07887860951
	450	-0.201875977	-0.1848964899	-0.07618140015
	500	-0.2213701537	-0.2112503316	-0.0734738041

Tableau 6.2 Valeurs de f^* des trois algorithmes pour $250 \leq n \leq 500$

Instances		$f^*_{Stats-Mads}$	f^*_{Mads}	f^*_{Gps}
PENALTY1 (Gould *et al.*, 2003)	250	4.473253553e+12	2.710362038e+13	2.710362038e+13
	300	1.412192327e+13	8.116802254e+13	8.116802254e+13
	350	2.60770444e+13	2.049837836e+14	2.049837836e+14
	400	4.541033039e+13	4.569066885e+14	4.569066885e+14
	450	1.053690702e+14	9.264901525e+14	9.264901525e+14
	500	1.761233361e+14	1.743373614e+15	1.743373614e+15
PENALTY2 (Gould *et al.*, 2003)	250	6.901666504e+17	9.446161685e+17	7.364529608e+17
	300	9.6129904e+21	1.043471597e+22	2.281953086e+22
	350	3.488646225e+26	4.582948854e+26	5.035068934e+26
	400	4.101084456e+30	1.10904776e+31	1.10904776e+31
	450	1.962295841e+35	2.442840255e+35	2.442840255e+35
	500	5.380713729e+39	5.380713732e+39	5.380713732e+39
POWELLSG (Gould *et al.*, 2003)	250	4709.022736	8671	8671
	300	5795.649361	11226	11346
	350	6732.021459	13686	13806
	400	7815.960503	16361	16481
	450	8679.669678	18821	18941
	500	11763.25	21616	21616
SROSENBER (Gould *et al.*, 2003)	250	68.71318309	1627	1627
	300	212.3480566	2093	2093
	350	895.3375	2559	2559
	400	554.6213281	3044	3044
	450	681.3408203	3510	3510
	500	1536.1425	3995	3995
VARDIM (Gould *et al.*, 2003)	250	279349251	271826933.6	416172313.7
	300	262187878	268078344.3	875516350.3
	350	4538393945	4528031116	1587463555
	400	4216304234	4206089786	2805595827
	450	4481006087	4488557510	4510159695
	500	4168725343	4023653988	6801558858
WOODS (Gould *et al.*, 2003)	250	875267.9686	994535	978611.8
	300	825991.7653	1228107.8	1223013.4
	350	1342458.848	1469240.6	1443128.6
	400	1520770	1706763	1692624.6
	450	1720014.8	1943946.2	1917834.2
	500	1944169.8	2162820.2	2162235.8

6.4 Extensions de STATS-MADS : résultats

Dans cette section, nous exposons les résultats des variantes de STATS-MADS que nous avons rencontré au chapitre 5.

STATS-MADS non itératif et STATS-MADS homogène

Les résultats de STATS-MADS non itératif et de STATS-MADS homogène sont fondés sur l'ensemble des 72 problèmes (tableaux 6.1 et 6.2).

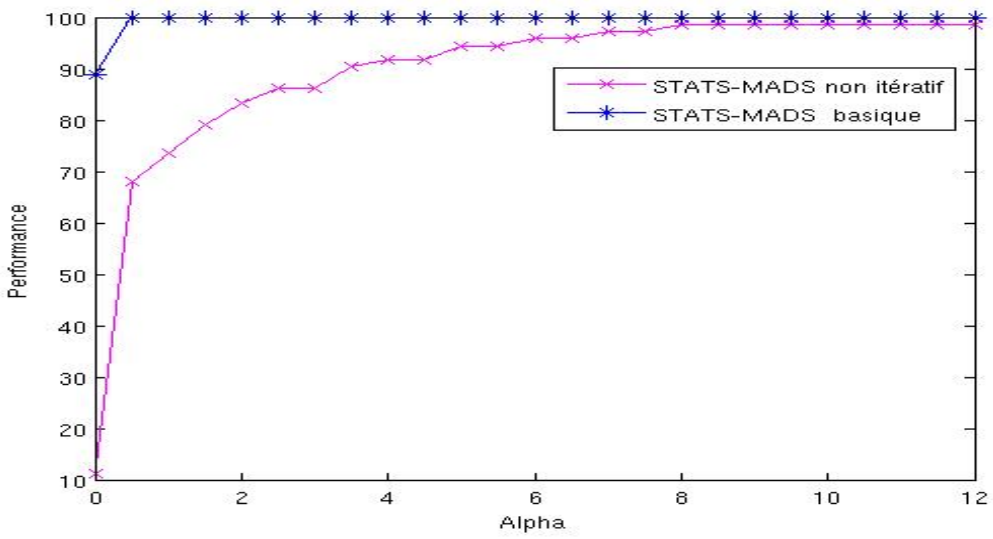

Figure 6.10 Comparaison de STATS-MADS non itératif et STATS-MADS basique, basée sur 72 problèmes

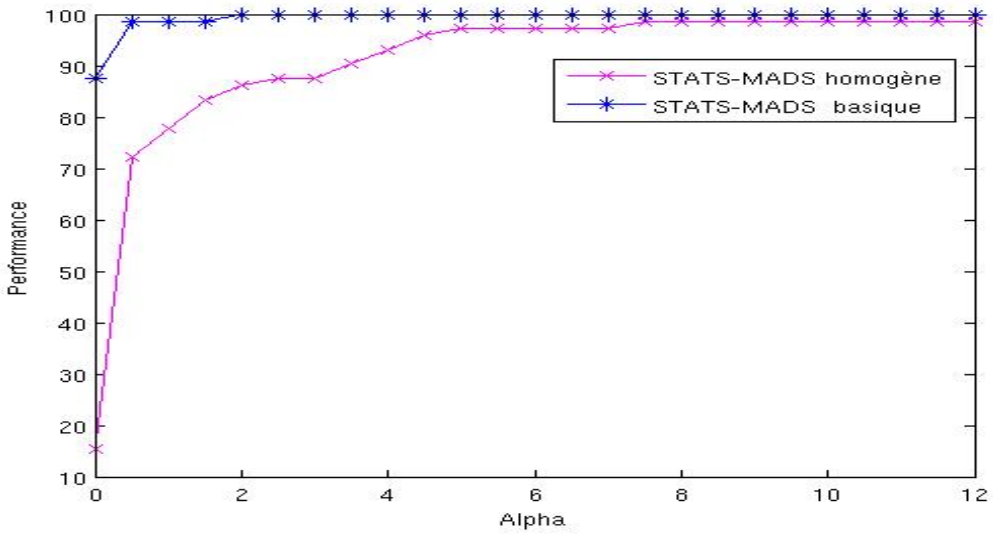

Figure 6.11 Comparaison de STATS-MADS homogène et STATS-MADS basique, basée sur 72 problèmes

Les profils de performance représentés par les figures 6.10 et 6.11 montrent que l'algorithme STATS-MADS est beaucoup plus performant que STATS-MADS homogène (où on fixe toujours les mêmes variables) et STATS-MADS non itératif (où on passe une seule fois à un sous-espace). Dans 90% des cas, il réussit à atteindre la valeur optimale f^*. Ces approches ne semblent donc pas être rentables en termes d'optimisation du nombre d'évaluations.

Méthodes de la frontière et des indices totaux

Nous testons les méthodes de la frontière et des indices totaux pour les problèmes de petite dimension : $n = 10$ et $n = 20$, car ces deux approches sont très gourmandes en temps de calcul pour $n > 20$. Nous obtenons les figures 6.12, 6.13, 6.14 et 6.15. Nous souhaitons comparer la figure 6.1 aux figures 6.12 et 6.14 et la figure 6.2 à 6.13

et 6.15 (ayant les mêmes dimensions et basés sur le même nombre de problèmes). Pour $n = 10$, la méthode des indices totaux favorise la fixation de 50% de variables et met GPS en deuxième rang, alors qu'elle garde STATS-MADS (90%) en troisième rang. Il est clair que le classement des variables n'est pas le même que STATS-MADS, mais le pourcentage de succès n'est toujours pas satisfaisant (inférieur à 50%). Pour $n = 20$, on obtient exactement le même classement d'algorithmes. Cependant, STATS-MADS (50%) obtient un taux de succès inférieur à 63.64% de la figure 6.2.

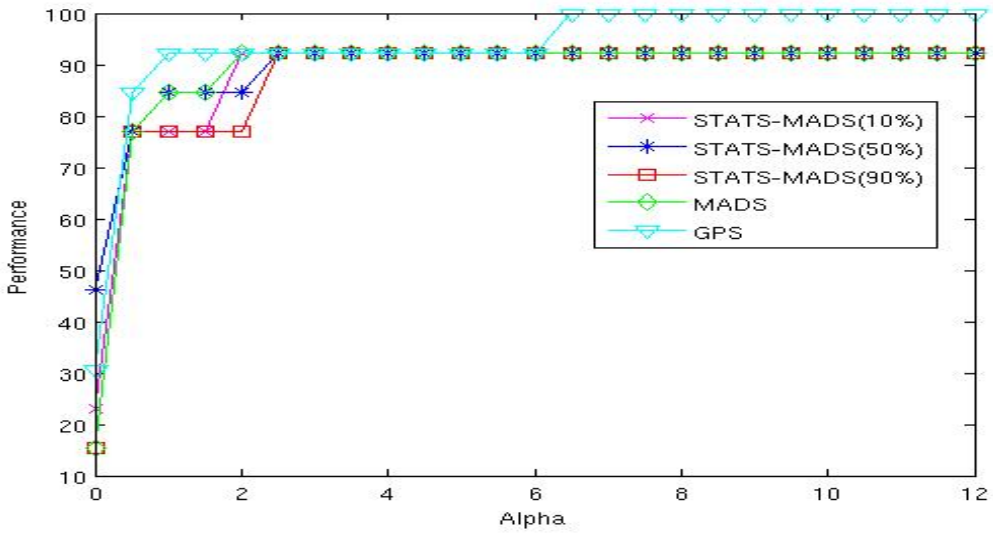

Figure 6.12 Profils de performance pour $n = 10$ en utilisant la méthode des indices totaux

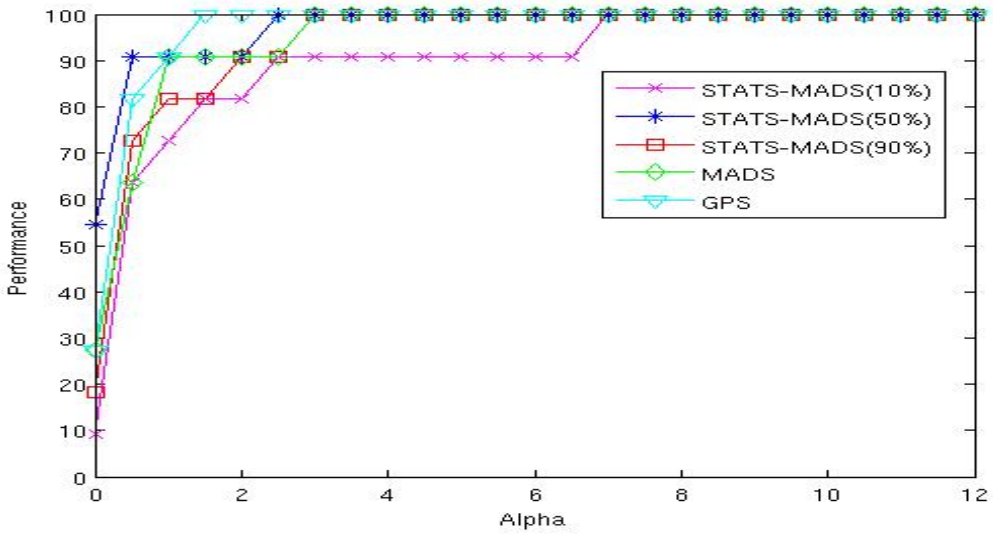

Figure 6.13 Profils de performance pour $n = 20$ en utilisant la méthode des indices totaux

Pour $n = 10$ (figure 6.14), la méthode de la frontière met en premier rang l'algorithme STATS-MADS (50%) avec un pourcentage de succès faible (38.46%), à égalité avec GPS et permute le classement de STATS-MADS (10%) et STATS-MADS (50%). Pour $n = 20$ (figure 6.15), le classement est le même pour les deux premières positions. Tout comme la méthode des indices totaux, nous percevons que STATS-MADS (50%) obtient un pourcentage inférieur (45.45%) à celui obtenu avec STATS-MADS basique (63.64%).

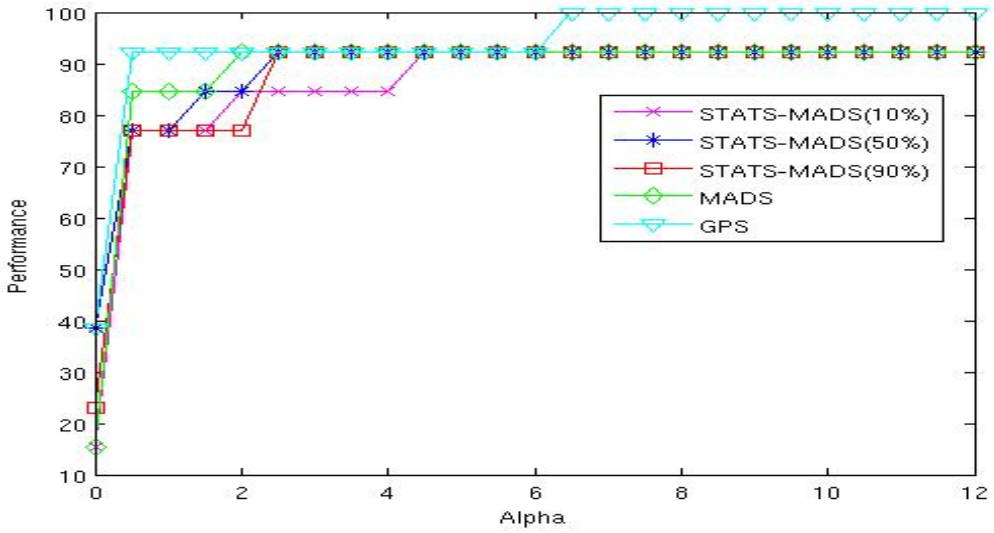

Figure 6.14 Profils de performance pour $n = 10$ en utilisant la méthode de la frontière

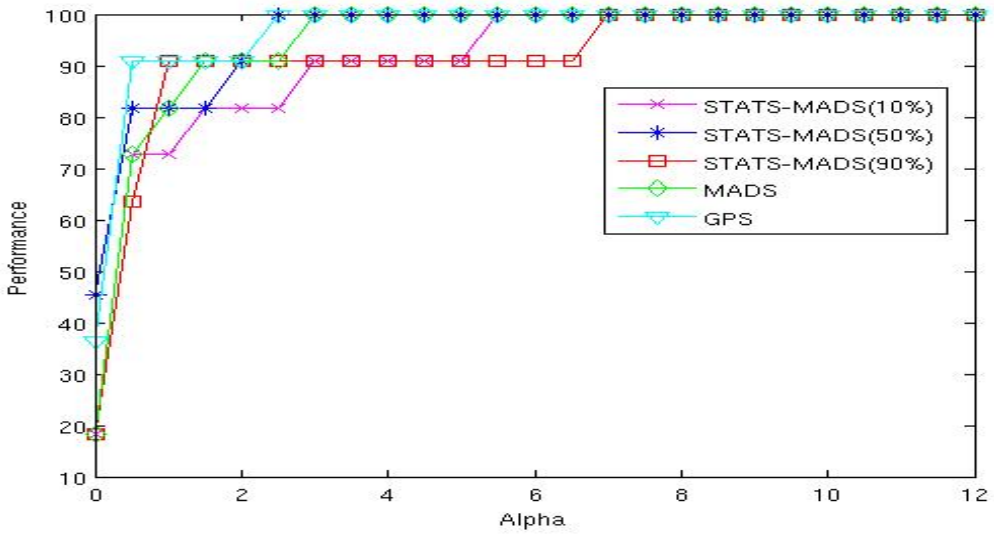

Figure 6.15 Profils de performance pour $n = 20$ en utilisant la méthode de la frontière

Il convient de ce qui précède qu'on ne peut rien conclure par rapport aux deux méthodes précédentes (une amélioration pour $n = 10$ en contre partie d'une détérioration pour $n = 20$). Le gain est à peine perceptible en petite dimension. Le temps de calcul devient

énorme si on passe à des dimensions plus importantes. Notons qu'une légère préférence est accordée à la méthode de la frontière, étant donné que le calcul des indices totaux est limité à un certain nombre de variables.

6.5 Autres résultats

Nous effectuons d'autres tests sur l'ensemble des 72 problèmes de dimension $250 \leq n \leq 500$ avec STATS-MADS. Nous rapportons les observations suivantes.

Comparaison avec une méthode aléatoire

Nous comparons STATS-MADS avec une méthode pour laquelle on fixe aléatoirement les variables dans les sous-espaces, comme dans PSD-MADS (Audet *et al.*, 2008c). Mise à part la technique de fixation, cette méthode est identique à STATS-MADS. Afin de ne pas fixer les mêmes variables, une graine aléatoire initialisant un générateur de nombres pseudo-aléatoires est utilisée. La figure 6.16 révèle l'utilité de l'approche statistique à déterminer les variables susceptibles d'accélérer le processus d'optimisation.

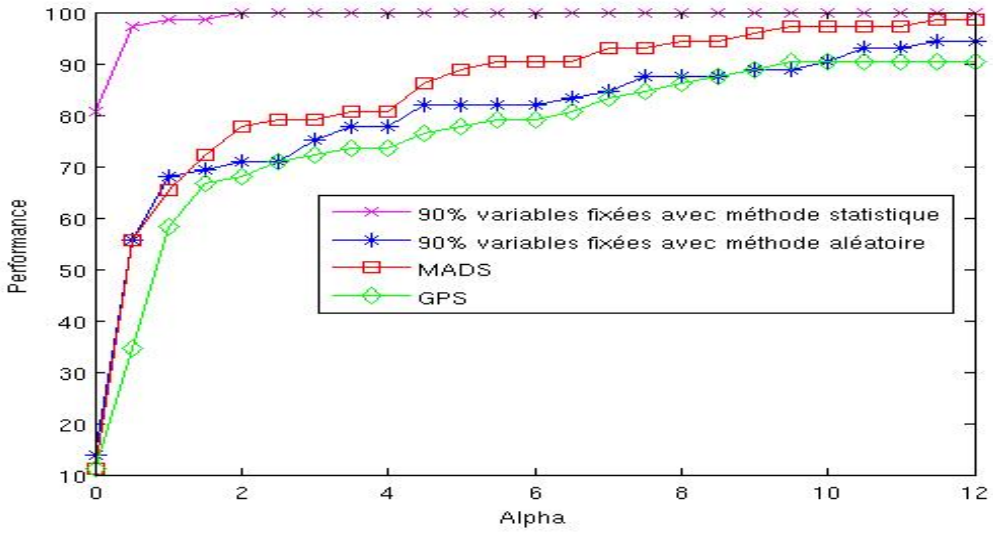

Figure 6.16 Profils de performance dans le cas où $250 \leq n \leq 500$ pour un ensemble de 72 problèmes avec la méthode aléatoire

Retour sur MADS-RESCUE

Nous avons remarqué, en phase de tests, que le recours au processus de secours MADS-RESCUE est beaucoup moins fréquent en grande dimension. Nous avons alors relancé les tests en exculant le MADS-RESCUE. Rappelons qu'il s'agit du processus noté P_3. L'alternance est alors limitée à P_1 et P_2. La figure 6.17 illustre ce comportement. Les résultats montrent que 66.67% de problèmes passent sans MADS-RESCUE en comparaison avec 61.11% . Les valeurs sont assez proches. 22.22% des problèmes donnent exactement le même résultat avec les deux stratégies. Il pourrait être alors judicieux de préconiser l'élimination de ce processus en grande dimension ($n \geq 250$). Cela peut être aussi intéressant du point de vue optimisation de paramètres, où il y aura deux paramètres de moins (M et τ) à définir.

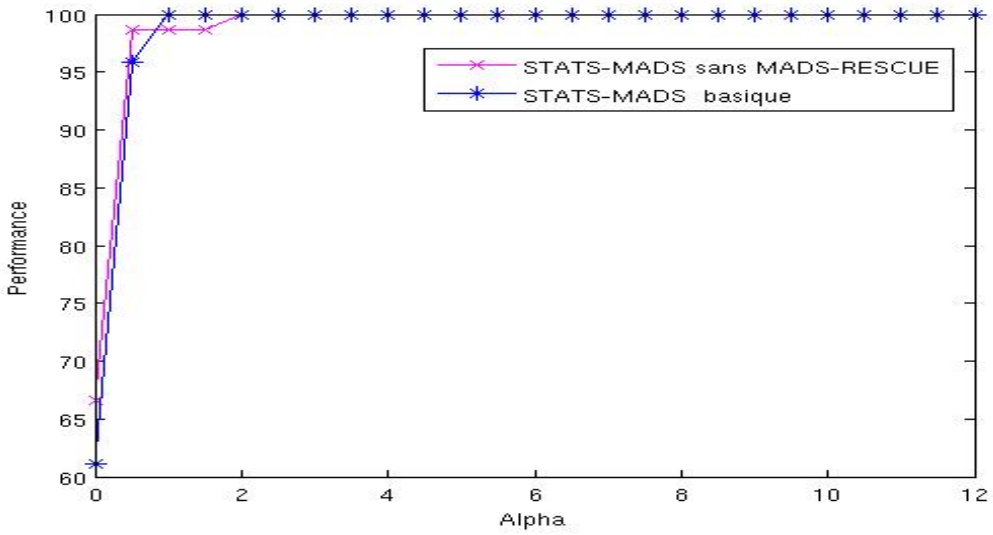

Figure 6.17 Comparaison entre STATS-MADS basique et STATS-MADS sans MADS-RESCUE

6.6 MOPTA08

Le problème test MOPTA08 ((Jones, 2008), (Anjos, 2009)) est un problème d'optimisation très complexe construit à partir d'un problème réel de General Motors qui consiste à la minimisation de la masse d'un véhicule motorisé sujette à des contraintes de performance. Il comporte $n = 124$ variables et 68 contraintes d'inégalité et a comme principal but la réduction de la fonction objectif en dessous de 228, à partir d'une valeur initiale égale à environ 251 en un nombre d'évaluations inférieur à 1800 ce qui correspond à peu près à 15 fois le nombre de variables. Cette réduction permettra d'optimiser le temps des simulations (une simulation prend approximativement 20 minutes). Afin d'atteindre cet objectif, plusieurs méthodes ont été utilisées. Celles-ci sont recensées et comparées dans (Langouët, 2011) et (Regis, 2011). Une recherche plus récente sur ce sujet est évoquée dans (Regis et Shoemaker, 2012).

L'algorithme MADS est loin d'aboutir à cet objectif. En effet, au bout de 1860 évaluations,

on obtient une valeur de fonction objectif aux alentours de 242 (dans les conditions standards). Le but de notre travail n'est pas de s'acharner à résoudre ce problème particulier, mais de mesurer l'impact de STATS-MADS par rapport à MADS.

Bien qu'il ne soit pas pertinent d'étudier ce problème de point de vue industriel, nous en exploitons les caractéristiques (complexité, nombre de variables, etc.) afin de tester l'algorithme STATS-MADS, conçu pour ce genre de problèmes. Nous nous limitons à la comparaison de ce dernier avec MADS et GPS, tel était le cas des problèmes tests précédents. Les paramètres de STATS-MADS ci-dessous ont été déterminés en fonction des tests :

- $n^p_{eval} = 20n$ pour P_1, $n^p_{eval} = 10n$ pour P_2 ;
- $n^{tot}_{eval} = 100n$;
- $M = 1000$;
- $\tau = 4$;
- $n_e = 1$;
- $pourcentage = 90$: le choix de ce paramètre est basé sur les recommandations précédentes.

Les profils de performance obtenus sont donnés par les figures 6.18 et 6.19. Pour la première, le critère d'arrêt est un budget sur le nombre d'évaluations égal à $100n$. Pour la deuxième, on lance chaque algorithme sur un nombre indéterminé d'évaluations, alors celui-ci s'arrête lorsque la taille du treillis devient plus petite que la précision de NOMAD. C'est ce qui explique que les trois algorithmes ne s'achèvent pas au bout du même nombre d'évaluations.

Les deux figures montrent, qu'avec les paramètres choisis, la descente locale est légèrement plus rapide dans le cas de STATS-MADS que MADS et GPS. Le gain en nombre d'évaluations, aussi petit soit-il, se traduit dans le milieu industriel par des gains de temps importants, étant donné le coût des simulations.

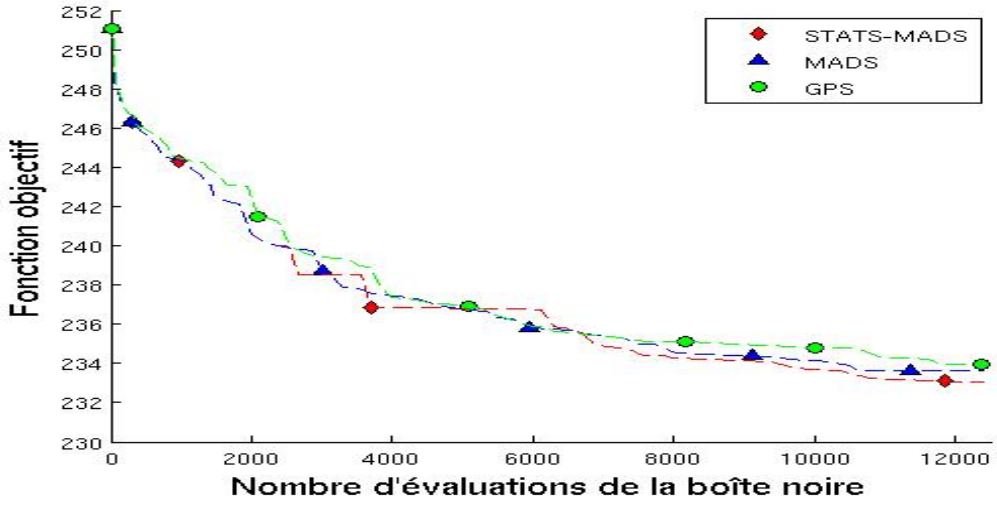

Figure 6.18 Profils de performance pour MOPTA08 : le critère d'arrêt est $100n$ évaluations

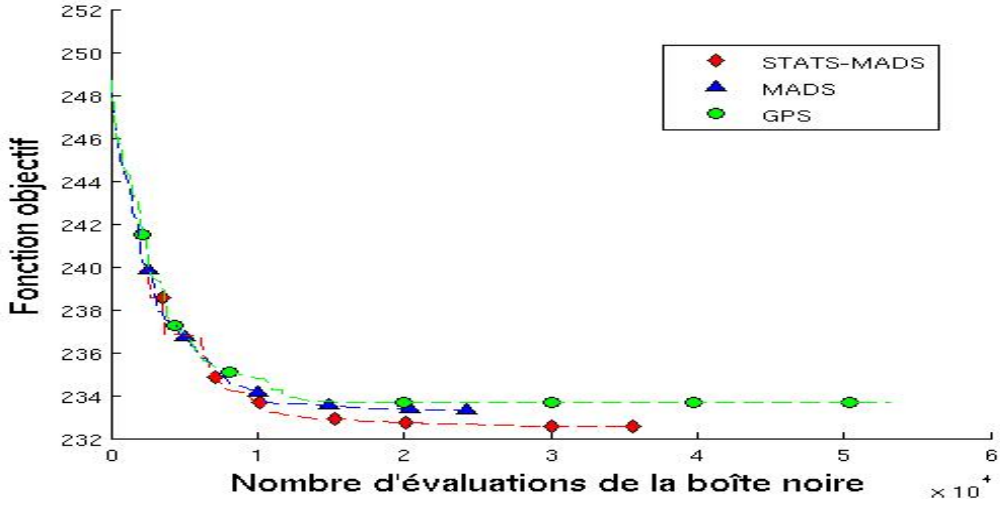

Figure 6.19 Profils de performance pour MOPTA08 : aucun critère d'arrêt n'est donné

Le chapitre suivant sera synthétique. Nous y discuterons les résultats obtenus. Nous y décriverons également les limites et les perspectives de recherche possibles de notre algorithme.

CHAPITRE 7

CONCLUSION

7.1 Synthèse des travaux

Un algorithme appelé STATS-MADS a été développé et implémenté en C++ sous NOMAD dont les principales finalités sont la réduction du nombre d'appels aux boîtes noires, des dispositifs à évaluations coûteuses et la gestion des problèmes de grande dimension. Afin d'accomplir cette tâche, plusieurs étapes ont été effectuées. D'abord, une méthode qui dérive de l'AS a été adoptée en fonction des spécifications du projet, permettant l'estimation des indices de sensibilité d'ordre un basé sur une égalité de l'ANOVA à un facteur et ceux d'ordre deux basé sur l'ANOVA factorielle. L'approximation des indices totaux a été alors possible en fonction des estimations précédentes. Ensuite, la structure générale de l'algorithme a été établie et ses différents ingrédients ont été définis. L'algorithme STATS-MADS résultant est une technique optimisant en alternance dans des sous-espaces distincts de variables et l'espace complet. Le passage d'un espace à un sous espace se fait au moyen d'une stratégie de fixation de variables les moins influentes identifiées par l'AS. STATS-MADS est guidé par trois processus dont deux correspondent à l'algorithme MADS classique duquel il hérite les propriétés de convergence. Des variantes de STATS-MADS ont également été proposées. Enfin, l'algorithme ainsi que ses extensions ont été testés sur un ensemble de problèmes numériques issus de la littérature et comparés à MADS et GPS.

7.2 Discussion générale

Nous avons proposé une manière de concevoir STATS-MADS et nous en avons testé diverses extensions. Nous sommes arrivés à la conclusion que STATS-MADS basique n'influe pas le temps de calcul et est le plus performant et le plus apte à atteindre notre objectif principal. En outre, l'accomplissement de celui-ci est mieux distingué de celui de MADS au fur et à mesure que la taille du problème augmente. Notre méthode s'est avérée plus efficace pour traiter les problèmes de grande dimension, plus précisément ceux dont la dimension est supérieure ou égale à 250. Ceci est favorable pour les problèmes industriels pour lesquels l'explosion dimensionnelle empêche l'utilisation de certains logiciels d'optimisation.

Nous ne pouvons trancher sur l'efficacité de notre méthode pour les plus faibles dimensions, étant donné que nous avons porté nos jugements sur la base d'un nombre insuffisant de problèmes testés.

La limitation principale de STATS-MADS est sa sensibilité au choix des paramètres. Nous recommandons au lecteur intéressé l'usage des paramètres définis à la phase des tests.

7.3 Perspectives de recherche

Deux perspectives de recherche peuvent être envisagées. D'une part, d'après (Audet et Orban, 2006) et (Audet *et al.*, 2010a) et (Dang, 2012), il est possible d'optimiser les paramètres d'un algorithme conçu à résoudre des problèmes d'optimisation. En effet, le choix de nos paramètres n'a été fondé que sur les problèmes tests. Il est certain que l'utilisation d'une stratégie algorithmique serait plus prometteuse.

D'autre part, il serait avantageux d'intéger notre méthode statistique dans PSD-MADS (Audet *et al.*, 2008c), où on cherche à résoudre des problèmes de grande dimension en

lançant des processus parallèles contenant des sous-ensembles de variables. Pour chaque processus, le choix des variables à fixer est aléatoire. Nous avons pu constater, lors des tests, l'intérêt de l'approche statistique par rapport à l'approche aléatoire.

RÉFÉRENCES

ABRAMSON, M. (2002). *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*. Thèse de doctorat, Department of Computational and Applied Mathematics, Rice University.

ABRAMSON, M. (2004). NOMADm optimization software. `http://www.afit.edu/en/enc/Faculty/MAbramson/nomadm.html`. Mis en ligne en 2004, consulté le 8 décembre 2011.

ABRAMSON, M. et AUDET, C. (2006). Convergence of mesh adaptive direct search to second-order stationary points. *SIAM Journal on Optimization*, 17(2):606–619.

ABRAMSON, M., AUDET, C., COUTURE, G., DENNIS, JR., J., LE DIGABEL, S. et TRIBES, C. (2012). The NOMAD project. Software available at `http://www.gerad.ca/nomad`.

ABRAMSON, M., AUDET, C., DENNIS, JR., J. et LE DIGABEL, S. (2009). OrthoMADS : A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, 20(2):948–966.

ALLARD, A., FISCHER, N., DIDIEUX, F., GUILLAUME, E. et IOOSS, B. (2011). Evaluation of the most influent input variables on quantities of interest in a fire simulation. *Journal de la société française de statistique*, 152(1):103–117.

ANDRES, T. H. et HAJAS, W. C. (1993). Using iterated fractional factorial design to screen parameters in sensitivity analysis of a probabilistic risk assessment model. *In* KÜSTERS, H., STEIN, E. et WERNER, W., éditeurs : *Proceedings of the Joint International Conference on Mathematical methods and Supercomputing in Nuclear Applications*, volume 2, pages 328–337.

ANJOS, M. (2009). Mopta 2008 benchmark. `http://anjos.mgi.polymtl.ca/MOPTA2008Benchmark.html`. Mis en ligne en 2009, consulté le 23 février 2012.

AUDET, C. (2004). Convergence Results for Pattern Search Algorithms are Tight. *Optimization and Engineering*, 5(2):101–122.

AUDET, C. (2011). A short proof on the cardinality of maximal positive bases. Rapport technique 1, Optimization Letters.

AUDET, C., BÉCHARD, V. et LE DIGABEL, S. (2008a). Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *Journal of Global Optimization*, 41(2): 299–318.

AUDET, C., CUSTÓDIO, A. et DENNIS, JR., J. (2008b). Erratum : Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 18(4):1501–1503.

AUDET, C., DANG, C.-K. et ORBAN, D. (2010a). Algorithmic parameter optimization of the dfo method with the opal framework. Rapport technique G-2010-02, Les cahiers du GERAD. To appear in Software Automatic Tuning, Springer.

AUDET, C. et DENNIS, JR., J. (2004). A pattern Search Filter Method for Nonlinear Programming without Derivatives. *SIAM Journal on Optimization*, 14(4):980–1010.

AUDET, C. et DENNIS, JR., J. (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217.

AUDET, C. et DENNIS, JR., J. (2009). A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, 20(4):445–472.

AUDET, C., DENNIS, JR., J. et LE DIGABEL, S. (2008c). Parallel space decomposition of the mesh adaptive direct search algorithm. *SIAM Journal on Optimization*, 19(3):1150–1170.

AUDET, C., DENNIS, JR., J. et LE DIGABEL, S. (2010b). Globalization strategies for mesh adaptive direct search. *Computational Optimization and Applications*, 46(2):193–215.

AUDET, C. et DENNIS, JR., J. E. (2003). Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903.

AUDET, C. et ORBAN, D. (2006). Finding optimal algorithmic parameters using derivative-free optimization. *SIAM Journal on Optimization*, 17(3):642–664.

BERGHEN, F. V. (2004). *CONDOR : A Constrained, Non-Linear, Derivative-Free Parallel Optimizer for Continuous, High Computing Load, Noisy Objective Functions*. Thèse de doctorat, Université Libre de Bruxelles, Belgium.

BETTONVIL, B. et KLEIJNEN, J. P. C. (1997). Searching for important factors in simulation models with many factors : sequential bifurcation. *European Journal of operational research*, 96:180–194.

BJÖRKMAN, M. et HOLMSTRÖM, K. (2000). Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering*, 1(4):373–397.

BOOKER, A., DENNIS, JR., J., FRANK, P., SERAFINI, D. et TORCZON, V. (1998). Optimization using surrogate objectives on a helicopter test example. *In* BORGGAARD, J., BURNS, J., CLIFF, E. et SCHRECK, S., éditeurs : *Optimal Design and Control*, Progress in Systems and Control Theory, pages 49–58, Cambridge, Massachusetts. Birkhäuser.

BOOKER, A. J., DENNIS, JR., J. E., FRANK, P. D., SERAFINI, D. B., TORCZON, V. et TROSSET, M. W. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13.

CHAN, C., SALTELLI, A. et TARANTOLA, S. (1997). Sensitivity analysis of model output : Variance-based methods make the difference. *In Proceedings of the 1997 Winter Simulation Conference*, pages 261–268, Italy. Environnement Institute European Commision Joint Research Centre.

CLARKE, F. H. (1983). *Optimization and Nonsmooth Analysis*. Wiley, New York. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

CONN, A., SCHEINBERG, K. et TOINT, P. (1998). A derivative free optimization algorithm in practice. *In Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4*.

CONN, A., SCHEINBERG, K. et VICENTE, L. (2006). Global convergence of general derivative-free trust-region algorithms to first and second order critical points. Rapport technique 06-49, Departamento de Matemática, Universidade de Coimbra, Portugal.

CONN, A., SCHEINBERG, K. et VICENTE, L. (2009). *Introduction to Derivative-Free Optimization*. MPS/SIAM Book Series on Optimization. SIAM, Philadelphia.

CONN, A. et TOINT, P. (1996). An algorithm using quadratic interpolation for unconstrained derivative free optimization. *In* PILLO, G. D. et GIANESSI, F., éditeurs : *Nonlinear Optimization and Applications*, pages 27–47. Plenum Publishing, New York.

CONN, A. R. et LE DIGABEL, S. (2011). Use of quadratic models with mesh adaptive direct search for constained black box optimization. Rapport technique G-2011-11, Les cahiers du GERAD.

COTTER, S. C. (1979). A screening design for factorial experiments with interactions. *Biometrika*, 66:317–320.

CUKIER, R. I., FORTUIN, C. M., SHULER, K. E., PETSCHEK, A. G. et SCHAIBLY, J. H. (1973). Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory. *chemical physics*, 59:3872–3878.

CUKIER, R. I., LEVINE, R. I. et SHULER, K. E. (1978). Nonlinear sensitivity analysis of multiparameter model systems. *Computational Physics*, 26:1–42.

CUKIER, R. I., SHULER, K. E. et SCHAIBLY, J. H. (1975). Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients-analysis of the approximations. *Chemical Physics*, 63:1140–1149.

DA-VEIGA, S. (2005). *Analyse d'incertitudes et de sensibilité : Application aux modèles de cinétique chimique*. Thèse de doctorat, Université de Toulouse III.

DANG, C. (2012). *Optimization of Algorithms with the Opal Framework*. Thèse de doctorat, École Polytechnique de Montréal.

DAVIS, C. (1954). Theory of positive linear dependence. *American Journal of Mathematics*, 76:733–746.

DE ROCQUIGNY, E., DEVICTOR, N. et TARANTOLA, S. (2008). *Uncertainty in Industrial Practice : A guide to quantitative uncertainty managmement*. Wiley.

DENNIS, J. E. et SCHNABEL, R. B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics, SIAM.

DENNIS, JR., J. et SCHNABEL, R. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ. Reissued in 1996 by SIAM Publications, Philadelphia, as Vol. 16 in the series Classics in Applied Mathematics.

DENNIS, JR., J. E. et TORCZON, V. (1991). Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474.

DOLAN, E. et MORÉ, J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

FERMI, E. et METROPOLIS, N. (1952). Los Alamos unclassified report LS–1492. Rapport technique, Los Alamos National Laboratory, Los Alamos, New Mexico.

FLETCHER, R. et LEYFFER, S. (2002). Nonlinear programming without a penalty function. *Mathematical Programming*, Series A, 91:239–269.

GILBERT, J. C. (1992). Automatic differentiation and iterative processes. *Optimization methods and software*, 1(1):13–21.

GOULD, N., ORBAN, D. et TOINT, P. (2003). CUTEr (and SifDec) : a constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394.

HALTON, J. (1960). On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals. *Numerische Mathematik*, 2(1):84–90.

HELTON, J. C. et DAVIS, F. J. (2002). Illustration of sampling-based methods for uncertainty and sensitivity analysis. *Risk Anal*, 22(3):591–622.

HOOKE, R. et JEEVES, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, 8(2):212–229.

IOOSS, B. (2011). Revue sur l'analyse de sensibilité globale de modèles numériques. *Journal de la Société Française de Statistique*, 152(1):3–25.

JACQUES, J. (2005). *Contributions à l'analyse de sensibilité et à l'analyse discriminante généralisée*. Thèse de doctorat, Université Joseph Fourier - Grenoble 1.

JEBALIA, M. (2008). *Optimisation par Stratégies d'Évolution : Convergence et vitesses de convergence pour des fonctions bruitées - Résolution d'un problème d'identification*. Thèse de doctorat, Université Pierre et Marie Curie.

JONES, D. (2008). Large-scale multi-disciplinary mass optimization in the auto industry. Presented at the MOPTA 2008 Conference (20 August 2008).

JONES, D. R., SCHONLAU, M. et WELCH, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

KIRKPATRICK, S., GELATT, C. D. et VECCHI, M. P. (1983). Optimization by simulated annealing. *Journal of Global Optimization*, 220(4598):671–680.

KOLDA, T. G., LEWIS, R. M. et TORCZON, V. (2003). Optimization by direct search : new perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482.

LANGOUËT, H. (2011). *Optimisation sans dérivées sous contraintes : deux applications industrielles en ingénierie de réservoir et en calibration des moteurs*. Thèse de doctorat, Université de Nice-Sophia Antipolis.

LE DIGABEL, S. (2009). NOMAD user guide. Rapport technique G-2009-37, Les cahiers du GERAD.

LE DIGABEL, S. (2011). Algorithm 909 : NOMAD : Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):1–15.

LEWIS, R. M. et TORCZON, V. (1996). Rank ordering and positive bases in pattern search algorithms. Rapport technique TR96-71, ICASE, NASA Langley Research Center.

LEWIS, R. M. et TORCZON, V. (1999). Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099.

LEWIS, R. M. et TORCZON, V. (2000). Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941.

LEWIS, R. M. et TORCZON, V. (2002). A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12:1075–1089.

LEWIS, R. M., TORCZON, V. et TROSSET, M. W. (2000). Direct search methods : Then and now. *Journal of Computational and Applied Mathematics*, 124(1–2):191–207.

LUKŠAN, L. et VLČEK, J. (2000). Test problems for nonsmooth unconstrained and linearly constrained optimization. Rapport technique V-798, ICS AS CR.

MATHWORKS, I. (2005). MATLAB GADS toolbox. `http://www.mathworks.com/products/gads/`.

MCKAY, M. (1997). Nonparametric variance-based methods of assessing uncertainty importance. *Reliability Engineering and System Safety*, 57:267–279.

MONTGOMERY, D. (2001). *Design and Analysis of experiments*. Wiley, 5th édition.

MORRIS, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33:161–174.

NELDER, J. A. et MEAD, R. (1965). A simplex method for function minimization. *Comput. J.*, 7:308–313.

POWELL, M. J. D. (2004). The NEWUOA software for unconstrained optimization without derivatives. Rapport technique DAMTP 2004/NA08, Department of Applied Mathematics and Theoretical Physics, University of Cambridge.

PRICE, C. et COOPE, I. (2003). Frame based ray search algorithms in unconstrained optimization. *Journal of Optimization Theory and Applications*, 116:259–377.

REGIS, R. G. (2011). Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers & Operations Research*, 38:837–853.

REGIS, R. G. et SHOEMAKER, C. A. (accepté, 2012). Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*.

ROSENBROCK, H. H. (1960). An automatic method for finding the greatest or least value of a function. *Comput. J.*, 3:175–184.

SALTELLI, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145:280–297.

SALTELLI, A., CHAN, K. et SCOTT, E. M. (2000). *Sensitivity analysis*. Wiley.

SALTELLI, A., RATTO, M., ANDRES, T., CAMPOLONGO, F., CARIBONI, J., GATELLI, D., SAISANA, M. et TARANTOLA, S. (2008). *Global sensitivity analysis - The Primer*. Wiley.

SALTELLI, A., TARANTOLA, S., CAMPOLONGO, F. et RATTO, M. (2004). *Sensitivity analysis in practice : A guide to Assessing Scientific Models*. Wiley.

SCHAIBLY, J. H. et SHULER, K. E. (1973). Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. ii applications. *Chemical Physics*, 59:3879–3888.

SOBOL, I. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers In Simulation*, 55:271–280.

TANG, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397.

TORCZON, V. (1991). On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1:123–145.

TORCZON, V. (1997). On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25.

TROSSET, M. (1997). I know it when I see it : Toward a definition of direct search methods. *SIAG/OPT Views-and-News : A Forum for the SIAM Activity Group on Optimization*, 9:7–10.

XU, C. et GERTNER, G. Z. (2008). Uncertainty and sensitivity analysis for models with correlated parameters. *Reliability Engineering and System Safety*, 93:1563–1573.

ANNEXE I

COMPLÉMENTS AU CHAPITRE 2

Définition I.1. *Une fonction f est dite Lipschitz en un point t s'il existe un scalaire K tel que pour tout y suffisamment proche de t :*

$$|f(t) - f(y)| \leq K\|t - y\|.$$

Définition I.2. *Une sous-suite d'optima locaux $\{x_k\}_{k\in K}$ du treillis est dite raffinante si $\left\{\Delta_k^p\right\}_{k\in K}$ tend vers 0.*

Définition I.3. *Si f est Lipschitz près de $\hat{x} \in \mathbb{R}^n$, alors la dérivée généralisée de Clarke en $\hat{x}$ dans la direction $d \in \mathbb{R}^n$ est :*

$$f^{\circ}(\hat{x}, d) = \lim_{y\to\hat{x}, t\downarrow 0} \sup \frac{f(y + td) - f(y)}{t}.$$

Définition I.4. *Un vecteur $v \in \mathbb{R}^n$ est dit hypertangeant au domaine $\Omega \subset \mathbb{R}^n$ au point $x \in \Omega$, s'il existe un scalaire $\epsilon > 0$ tel que : $y + tw \in \Omega$, $\forall y \in \Omega \cup B_\epsilon(v)$, $\forall w \in B_\epsilon(v)$ et $\forall 0 < t < \epsilon$, où : $B_\epsilon(x)$ est la boule de rayon ϵ centrée en x. L'ensemble $T_\Omega^H(x)$ des vecteurs tangents est appelé cône hypertangent en x.*

Définition I.5. *Un vecteur $v \in \mathbb{R}^n$ est dit Clarke-tangeant au domaine fermé $\Omega \subset \mathbb{R}^n$ au point $x \in \Omega$, si pour toute séquence $\{y_k\} \in \Omega$ convergeant vers x et si pour toute séquence $\{t_k\} > 0 \in \mathbb{R}$ convergeant vers 0, il existe une séquence de vecteurs $\{w_k\}$ convergeant vers v telle que $y_k + t_k w_k \in \Omega$. L'ensemble $T_\Omega^{Cl}(x)$ des vecteurs Clarke-tangeants est appelé cône de Clarke en x.*

Définition I.6. *Un vecteur $v \in \mathbb{R}^n$ est dit contigent au domaine fermé $\Omega \subset \mathbb{R}^n$ au point $x \in \Omega$, s'il existe une séquence $\{y_k\} \in \Omega$ convergeant vers x et s'il existe une séquence $\{\lambda_k\} > 0 \in \mathbb{R}$ telle que $v = \lim_k \lambda_k(y_k - x)$. L'ensemble $T_\Omega^B(x)$ des vecteurs contingents est appelé cône de Bouligand en x.*

Définition I.7. *Un ensemble est dit régulier en x lorsque $T_\Omega^{Cl}(x) = T_\Omega^B(x)$.*