Titre: Title:	Quantification de l'incertitude avec un ensemble de substituts pour l'optimisation de boîtes noires
Auteur: Author:	Renaud Saltet
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Saltet, R. (2021). Quantification de l'incertitude avec un ensemble de substituts pour l'optimisation de boîtes noires [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9104/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/9104/
Directeurs de recherche: Advisors:	Charles Audet
Programme: Program:	Maîtrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Quantification de l'incertitude avec un ensemble de substituts pour l'optimisation de boîtes noires

RENAUD SALTET
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Août 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Quantification de l'incertitude avec un ensemble de substituts pour l'optimisation de boîtes noires

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

Jonathan JALBERT, président
Charles AUDET, membre et directeur de recherche
Carolina OSORIO, membre

REMERCIEMENTS

Tout d'abord, je remercie vivement Charles Audet et Sébastien Le Digabel, professeurs à Polytechnique Montréal, pour m'avoir accueilli dans leur équipe et avoir dirigé ce mémoire. Leur expertise, leur disponibilité mais aussi leur confiance et leurs encouragements m'ont donné les moyens de mener à bien ce travail.

Je tiens à remercier Juliane Müller, chercheure au Laboratoire national Lawrence Berkeley, pour m'avoir fait découvrir son domaine de recherche et m'avoir mis en contact avec cette équipe fameuse du GERAD.

Je remercie sincèrement Viviane Rochon Montplaisir et Christophe Tribes, associés de recherche à Polytechnique Montréal, pour leurs nombreux conseils techniques et leur patience sans lesquels ce travail aurait été bien plus long et difficile.

Je tiens également à exprimer ma gratitude et ma sympathie envers les autres étudiants du GERAD pour m'avoir chaleureusement accueilli durant les toutes premières semaines de ma maîtrise. L'année 2020 ne nous aura malheureusement pas permis de faire plus ample connaissance.

Je remercie bien affectueusement mes parents Viviane et Jean Saltet. Leur soutien absolu, leur sollicitude et leur admiration m'ont donné le souffle pour réussir toutes ces années d'études, que le présent mémoire vient couronner.

Enfin, je remercie Clara Ducher, qui partage ma vie, pour son soutien moral et sa prévenance. Nos longues discussions m'ont permis d'enrichir mes travaux, d'y voir clair et de tenir bon.

RÉSUMÉ

En optimisation sans dérivées et de boîtes noires, les fonctions définissant le problème n'ont pas de forme analytique connue. Les dérivées ne sont notamment pas accessibles, ce qui rend caduc un grand nombre d'algorithmes existants. Une méthode éprouvée en optimisation de boîtes noires consiste à construire des modèles de l'objectif et, le cas échéant, des contraintes, grâce aux valeurs déjà connues. Ces modèles, peu coûteux en temps de calcul comparés aux vraies fonctions, permettent de guider l'optimisation puisqu'ils procurent une information sur le comportement du vrai problème.

Une technique connue consiste à utiliser pour une même fonction non pas un seul modèle mais un ensemble de modèles afin de tirer avantage de chacun d'entre eux. Ce mémoire propose une extension aux ensembles de modèles leur permettant de fournir en tout point de l'espace non seulement une prédiction mais aussi une mesure d'incertitude. Les ensembles de modèles ainsi modifiés se comportent comme des modèles stochastiques, ce qui permet d'utiliser des outils de l'optimisation bayésienne. La mesure d'incertitude proposée repose sur les différences de variations entre les modèles, et se décline en deux versions : une pour l'objectif et une pour les contraintes.

Pour tester cette approche, de tels ensembles de modèles ont été intégrés dans l'algorithme de recherche directe sur treillis adaptatifs (MADS). À chaque itération, un sous-problème issu de l'optimisation bayésienne et faisant intervenir les ensembles de modèles est résolu afin de trouver des points candidats. Le but est de vérifier que l'approche proposée améliore le choix de points par rapport aux méthodes existantes et en particulier par rapport aux modèles stochastiques.

La version de l'algorithme MADS ainsi conçue a été testée sur des problèmes variés : sept problèmes analytiques, deux problèmes d'optimisation multi-disciplinaire et deux problèmes de simulation. L'approche proposée a été comparée à deux autres versions de MADS : une sans étape de recherche globale et une avec des modèles quadratiques ; ainsi qu'à deux autres solveurs d'optimisation de boîtes noires. Les résultats montrent que notre approche est préférable aux autres versions de MADS et aux autres solveurs sur la plupart des problèmes difficiles. De plus, les modèles agrégés conçus sont plus rapides et plus précis que les modèles stochastiques sur les problèmes où ces derniers ont été testés. La méthode proposée est en revanche surpassée sur les problèmes analytiques et sur un problème de simulation dont la plupart des contraintes sont linéaires. Par ailleurs, un des deux solveurs concurrents n'est compétitif avec aucune version de MADS. L'autre solveur produit de bons résultats mais

nécessite un temps de calcul bien supérieur, et sur la plupart des problèmes il est préférable d'exécuter plusieurs fois l'algorithme proposé.

Étant donnés ces résultats, on peut conclure que les ensembles de modèles construits améliorent les résultats de l'algorithme MADS de référence et sont par ailleurs une alternative sérieuse aux modèles stochastiques.

ABSTRACT

In derivative-free and blackbox optimization, the analytical expression of the problem is unknown. In particular, no derivatives can be used and as a consequence many optimization algorithms are inoperative. In this context, an efficient method is to build models of the objective and constraints of the problem with the values that are already known. These models are cheap to compute compared to the actual functions and enable to guide the optimization since they bring information on the behaviour of the true problem.

This master thesis proposes an extension to ensembles of models, a technique that consists in using several models of the same function simultaneously. The extended ensembles of models provide at any point not only a prediction, but also an uncertainty on that prediction. They therefore behave like stochastic models, thus enabling to use efficient tools from Bayesian optimization. The proposed measure of uncertainty is based on the differences in the variations of the models and is given in two versions : one for the objective and one for the constraints.

In order to test the approach, the extended aggregate models have been integrated into the MADS algorithm. At each iteration, a subproblem inspired by Bayesian optimization is solved to find candidate points on which the true problem is evaluated. The purpose is to test whether the proposed approach improves the selection of candidate points compared to existing versions of MADS and especially to stochastic models.

The resulting MADS algorithm instance has been tested on various problems : seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. The proposed approach has been compared to two other versions of MADS : one without any search step and one with a search involving the minimization of quadratic models; as well as two other blackbox optimization solvers. The results show that the proposed approach outperforms the other versions of MADS and the other solvers on most of the difficult problems. Moreover, the extended aggregate models are faster and more accurate than the stochastic models on all the problems where the latter have been tested. However, the proposed method is outperformed on the analytical problems and on one simulation problem where most of the constraints are linear. Furthermore, one of the two competing solvers is not competitive with any version of MADS, and the other solver takes a large amount of time which is better invested running the proposed method multiple times.

Based on these results, the extended aggregate models improve the performance of the base MADS algorithm and constitute an alternative to stochastic models.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

DFO	Optimisation sans dérivées (*Derivative Free Optimization*)
BBO	Optimisation de boîtes noires (*Blackbox Optimization*)
CS	Recherche par coordonnées (*Coordinate Search*)
GPS	Recherche par motifs généralisée (*Generalized Pattern Search*)
MADS	Recherche directe par treillis adaptatifs (*Mesh Adaptive Direct Search*)
LtMADS	*Lower Triangular MADS*
OrthoMADS	*Orthogonal MADS*
NOMAD	*Nonlinear Optimization by Mesh Adaptive Direct Search*
EB	Barrière extrême (*Extreme Barrier*)
PB	Barrière progressive (*Progressive Barrier*)
EI	*Expected Improvement*
P	*Probability of Feasibility*
PI	*Probability of Improvement*
DFN	*Derivative-Free Nonsmooth*
SHEBO	*Surrogate optimization of problems with Hidden constraints and Expensive Black-box Objectives*

LISTE DES NOTATIONS

f	Fonction objectif
$\mathcal{X}$	Domaine de la fonction objectif
Ω	Ensemble réalisable
n	Nombre de variables
$\mathbb{R}$	Ensemble des nombres réels
$\overline{\mathbb{R}}$	$\mathbb{R} \cup \{-\infty, +\infty\}$
m	Nombre de contraintes relaxables
c_j	Contrainte relaxable numéro j
h	Fonction de violation des contraintes
s	Nombre de substituts
$\tilde{f}^p$	Substitut numéro p de la fonction objectif f
w^p	Poids affecté au substitut $\tilde{f}^p$
$\hat{f}$	Modèle agrégé de la fonction objectif f
$\tilde{c}_j^p$	Substitut numéro p de la contrainte c_j
w_j^p	Poids affecté au substitut $\tilde{c}_j^p$
$\hat{c}_j$	Modèle agrégé de la contrainte c_j
k	Compteur d'itérations
x^0	Solution initiale
x^k	Solution courante à l'itération k
δ^k	Taille du treillis à l'itération k
Δ^k	Taille du cadre de sonde à l'itération k
S^k	Ensemble des directions de sonde à l'itération k
M^k	Ensemble des points du treillis à l'itération k
P^k	Ensemble des points de sonde à l'itération k
V^k	Cache à l'itération k
$\mathbb{X}$	Ensemble d'échantillonnage
$\|\|x\|\|_\infty$	Norme infinie de x

CHAPITRE 1 INTRODUCTION

On considère un problème d'optimisation de la forme :

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x) \\ \text{s.c.} \quad & x \in \Omega \end{aligned} \tag{P}$$

où $\mathcal{X}$ est un sous-ensemble de $\mathbb{R}^n$; f est une fonction à variables dans $\mathbb{R}^n$ et à valeurs dans $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ appelée *fonction objectif*, ou simplement *objectif* ; Ω est l'ensemble $\Big\{x \in \mathcal{X} \mid c_j(x) \leq 0,\ 1 \leq j \leq m\Big\}$, appelé *ensemble réalisable* ; et les c_j sont des fonctions à variables dans $\mathbb{R}^n$ et à valeurs dans $\overline{\mathbb{R}}$ appelées *contraintes*. L'ensemble $\mathcal{X}$ contient les points qui respectent les contraintes dites *non relaxables*. Tous les points explorés au cours de l'optimisation doivent se trouver dans cet ensemble, soit parce que la fonction f n'est pas définie en dehors, soit parce qu'un point en dehors n'a pas de sens dans le problème de départ, *e.g.*, une longueur ne peut pas être négative. L'ensemble $\mathcal{X}$ est typiquement formé par des contraintes de bornes, c'est-à-dire que $\mathcal{X}$ est de la forme $\mathcal{X} = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ où ℓ et u sont des vecteurs de $\{\mathbb{R} \cup \{-\infty\}\}^n$ et $\{\mathbb{R} \cup \{+\infty\}\}^n$ respectivement. Les fonctions c_j sont des contraintes dites *relaxables*, c'est-à-dire qu'elles peuvent être violées au cours de l'optimisation. Toutefois, la solution finale doit satisfaire ces contraintes.

1.1 Optimisation de boîtes noires

En optimisation de boîtes noires (*Blackbox Optimization*) (BBO), on fait l'hypothèse que la forme analytique de l'objectif f et des contraintes c_j n'est pas connue. On n'a accès à aucune information au-delà des valeurs que prennent ces fonctions, ce qui empêche par exemple d'exploiter leurs dérivées, qui peuvent même ne pas exister. C'est pourquoi on qualifie ces fonctions de *boîtes noires*. Ce contexte apparaît typiquement lorsque le résultat de la fonction objectif est issu d'un code informatique. Il peut aussi apparaître si les valeurs de l'objectif sont le résultat d'une expérience en laboratoire. Une boîte noire est également coûteuse à évaluer : l'obtention d'une valeur peut prendre plusieurs secondes [24], plusieurs minutes [7, 62], plusieurs heures [18, 83], voire plusieurs jours [54]. L'optimisation de boîtes noires consiste à concevoir des algorithmes capables de trouver la meilleure solution possible à ce genre de problèmes avec un temps de calcul raisonnable, c'est-à-dire avec un certain budget d'évaluations à ne pas dépasser.

Au premier abord, l'optimisation de boîte noire s'apparente à l'optimisation sans dérivées

(*Derivative Free Optimization*) (DFO), qui, comme son nom l'indique, est l'étude des algorithmes d'optimisation qui n'utilisent pas les dérivées des fonctions définissant le problème. Il y a toutefois une nuance entre ces deux disciplines. Audet et Hare [14] les distinguent de la façon suivante : la DFO fait l'hypothèse que, bien qu'elles ne soient pas accessibles, les dérivées existent ; alors que la BBO ne fait aucune hypothèse sur l'existence des dérivées. Ainsi, la DFO est l'étude mathématique des algorithmes d'optimisation qui n'utilisent pas les dérivées mais dont les résultats de convergence [1] reposent sur leur existence. La BBO consiste quant à elle à concevoir des algorithmes efficaces pour des problèmes qui peuvent ne pas avoir de dérivées, sans nécessairement se soucier de résultats de convergence. Cette dernière discipline inclue donc les heuristiques, c'est-à-dire les méthodes qui n'ont pas de garantie de convergence mais qui peuvent être efficaces en pratique. Pour une meilleure compréhension de l'intérêt théorique de l'existence des dérivées en DFO, voir [15].

Il existe deux livres de référence en optimisation sans dérivées et de boîtes noires : *Introduction to Derivative-Free Optimization* de Conn, Scheinberg et Vicente [27], et *Derivative-Free and Blackbox Optimization* de Audet et Hare [14].

1.2 Contexte et objectif de la recherche

Une technique bien établie en BBO consiste à recourir à des substituts des fonctions du problème. Ces substituts sont supposés peu coûteux en temps de calcul comparés aux vraies fonctions, et par conséquent leur utilisation permet de guider l'optimisation tout en économisant des évaluations. Dans ce mémoire, les substituts employés sont des modèles de régression ou d'interpolation, qui seront plutôt appelés des *modèles*. Dans le cadre de ce travail, deux types de modèles importent : les modèles stochastiques et les ensembles de modèles. Avec les premiers, l'objectif est interprété comme un processus aléatoire. En conséquence, un modèle stochastique est ajusté aux valeurs connues et est utilisé pour fournir en tout point de l'espace non seulement une prédiction mais aussi une mesure de l'incertitude sur cette prédiction. Comme cette incertitude est généralement élevée dans les zones où les valeurs de l'objectif sont peu connues, elle peut être utilisée pour réaliser un compromis entre exploration de l'espace et intensification de la recherche dans les zones prometteuses. Cette approche porte aussi le nom d'optimisation bayésienne et a donné lieu à de nombreux développements. L'autre approche consiste à utiliser un ensemble de modèles pour un même objectif. Elle est motivée par le constat qu'il n'existe pas un seul modèle dominant, c'est-à-dire meilleur pour

1. On entend par *résultat de convergence* une garantie théorique que, lorsque le nombre d'itérations de l'algorithme est suffisamment grand, la solution produite se trouve arbitrairement proche d'un point satisfaisant des conditions nécessaires d'optimalité telles que $\nabla f(x) = 0$.

tous les problèmes. Par conséquent, utiliser plusieurs modèles en même temps permet de bénéficier des avantages de chacun. Les deux approches susnommées sont distinctes et n'ont *a priori* pas de lien entre elles.

Ce mémoire propose une extension des ensembles de modèles leur permettant de fournir en tout point non seulement une prédiction mais aussi une incertitude. Ainsi, un ensemble de modèles se comporte comme un modèle stochastique, ce qui permet d'employer des outils de l'optimisation bayésienne. Bien qu'ils ressemblent aux modèles stochastiques, on s'attend à ce que les nouveaux ensembles de modèles mettent en valeur des zones de l'espace de recherche différentes. On espère aussi qu'ils constituent une alternative moins coûteuse en temps de calcul, les modèles stochastiques étant parfois très longs à construire ou mettre à jour.

1.3 Plan du mémoire

Ce mémoire par article est organisé de la façon suivante. Le chapitre 2 propose une revue de littérature étendue sur les méthodes de l'optimisation de boîtes noires. Le chapitre 3 explique la démarche du travail par rapport à l'état de l'art. Le chapitre 4 présente la contribution sous la forme d'un article soumis au journal *Computational Optimization and Applications*. Le chapitre 6 fournit une synthèse des travaux ainsi que des pistes pour de futurs développements.

CHAPITRE 2 REVUE DE LITTÉRATURE

Les méthodes d'optimisation de boîtes noires peuvent être classées en trois catégories : les heuristiques, les méthodes de recherche directe, et les méthodes utilisant des substituts. La première catégorie désigne les méthodes pour lesquelles il n'existe pas de résultat de convergence, mais dont le développement est utile puisque les résultats peuvent être bons en pratique. C'est le cas des algorithmes évolutionnaires [14, Chapitre 4] et de l'algorithme de Nelder-Mead [64]. Les méthodes évolutionnaires sont peu efficaces en BBO car elles requièrent un grand nombre d'évaluations, ce que l'on cherche précisément à éviter. L'algorithme de Nelder-Mead est en revanche très utilisé en pratique car il s'avère efficace pour trouver un optimum local pour beaucoup de problèmes. Il reste néanmoins, dans sa première version, une heuristique puisqu'il existe des problèmes pour lesquels l'algorithme converge vers un point non stationnaire, c'est-à-dire un point où le gradient de l'objectif n'est pas nul [56]. La deuxième catégorie regroupe les méthodes dites de recherche directe dans lesquelles on procède en comparant la valeur de l'objectif sur un ensemble de points candidats à la solution courante. Ces méthodes font l'objet de la section 2.1. La troisième et dernière catégorie désigne les méthodes qui utilisent des substituts de l'objectif ou des contraintes censés imiter leur comportement tout en étant moins coûteux. Ces méthodes font l'objet de la section 2.2. Les méthodes heuristiques ne seront pas décrites davantage dans cette revue de littérature car nous n'en ferons pas usage.

La classification qui vient d'être faite est schématique. Les méthodes développées concrètement empruntent plusieurs aspects à chacune des trois catégories. En particulier une méthode de recherche directe intègre souvent une heuristique ou utilise des substituts afin d'améliorer ses résultats. Cette classification est toutefois utile pour comprendre l'esprit des méthodes de BBO car aucune des trois catégories n'est réductible aux autres.

2.1 Méthodes de recherche directe

Les méthodes de recherche directe peuvent être décrites comme les algorithmes fondés sur les valeurs de la fonction objectif et les comparaisons entre elles. Bien que la définition du terme *recherche directe* ne soit pas unique, nous retiendrons celle que Hooke et Jeeves [42] ont proposée en 1961 :

> Nous utilisons le terme « recherche directe » pour décrire l'évaluation consécutive de points candidats et la comparaison de ces points avec la « meilleure » solution

obtenue jusqu'à présent, avec une stratégie pour déterminer les futurs points candidats en fonction des résultats obtenus.

Nous allons présenter dans cet ordre les méthodes de recherche par coordonnées (*Coordinate Search*) (CS) [33], de recherche par motifs généralisée (*Generalized Pattern Search*) (GPS) [82] et de *Mesh Adaptive Direct Search* (MADS) [10]. Chacune d'entre elle est une généralisation de la précédente. Elles ne sont pas des algorithmes prêt à l'emploi mais plutôt des cadres algorithmiques bénéficiant de propriétés de convergence et offrant une certaine liberté dans l'implémentation.

Pour une revue de littérature sur les méthodes de recherche directe, voir [6] et [50, Section 2.2]. Pour une revue des applications, voir [4].

2.1.1 Recherche par coordonnées (CS)

L'algorithme CS fut introduit en 1952 par Fermi en Metropolis [33]. Cette méthode simple est le fondement d'algorithmes plus sophistiqués qui font aujourd'hui partie de l'état de l'art. De plus, il possède des résultats de convergence sous certaines hypothèses.

Soit un point initial $x^0 \in \mathbb{R}^n$ et un pas initial $\delta^0 \in \mathbb{R}$ strictement positif. On note x^k la solution courante à l'itération $k \geq 1$, c'est-à-dire la meilleure solution connue.

À chaque itération, $2n$ points candidats sont créés en se déplaçant par rapport à la solution courante x^k dans la direction de chaque coordonnée tour à tour d'un pas δ^k. L'ensemble des $2n$ points ainsi créés s'écrit

$$P^k = \left\{ x^k \pm \delta^k e_i \ : \ 1 \leq i \leq n \right\} \tag{2.1}$$

où n et la dimension du problème et e_i est le vecteur dont le i-ième élément vaut 1 et les autres 0. On procède ensuite à l'étape de *sonde*. La fonction objectif est évaluée sur l'ensemble P^k, ce qui donne lieu à deux possibilités. Soit il existe un point y de P^k qui est meilleur que x^k, *i.e.*, $f(y) < f(x^k)$, auquel cas y devient la solution courante à l'itération suivante et δ^k ne change pas. On dit alors que l'itération est un *succès*. Soit aucun point de P^k n'est meilleur que x^k, auquel cas x^k reste la solution courante et le pas δ^k est divisé par deux à l'itération suivante. On dit alors que l'itération est un *échec*. L'idée est que si aucun point autour de x^k n'est meilleur, alors x^k est probablement proche d'un optimum et il faut par conséquent affiner la recherche autour de lui en diminuant le pas. On s'arrête lorsque δ^k est inférieur à un certain seuil ϵ_{stop} donné au départ. Cette procédure est formalisée dans l'algorithme 1.

Algorithme 1 : Recherche par coordonnées (CS)

0. Initialisation :

$x^0 \in \mathbb{R}^n$: point initial
$\delta^0 > 0$: pas initial
$\epsilon_{\text{stop}} > 0$: critère d'arrêt
$k \leftarrow 0$: compteur d'itérations

1. Sonde :

si $\exists\ y \in P^k,\ f(y) < f(x^k)$ **alors**
  $x^{k+1} \leftarrow y$
  $\delta^{k+1} \leftarrow \delta^k$
sinon
  $x^{k+1} \leftarrow x^k$
  $\delta^{k+1} \leftarrow \delta^k/2$

2. Critère d'arrêt :

si $\delta^{k+1} \geq \epsilon_{\text{stop}}$ **alors**
  $k \leftarrow k+1$
  Aller à 1
sinon
  STOP

Les propriétés de convergence de l'algorithme CS se résument au théorème 2.1 [82] traduit de [14, Chapitre 3].

Théorème 2.1 (Convergence de la recherche par coordonnées) *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction de classe $\mathcal{C}^0$ (i.e., continue) dont les ensembles de niveau sont bornés et $\{x^k\}$ la suite des itérés engendrée par CS avec comme critère d'arrêt $\epsilon_{\text{stop}} = 0$. Soit $\hat{x}$ un point d'accumulation des itérés ayant conduit à un échec de $\{x^k\}$. Alors pour toute direction $d \in \left\{ \pm\ e_i\ : 1 \leq i \leq n \right\}$, soit $f'(\hat{x}; d) \geq 0$, soit $f'(\hat{x}; d)$ n'existe pas. De plus, si f est de classe $\mathcal{C}^1$, alors $\hat{x}$ est un point critique, i.e., $\nabla f(\hat{x}) = 0$.*

Ce résultat montre que l'on peut garantir la non-décroissance de f dans les directions des coordonnées. Audet et Hare [14, Chapitre 3] donnent un exemple pour lequel l'algorithme CS ne donne pas de solution satisfaisante malgré le théorème 2.1. Il suffit de prendre la fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie par $f(x) = ||x||_\infty$ et le point initial $x = [1, 1]^\top$. Alors pour toute valeur de δ, le point $x^0 \pm \delta e_i$ ($i \in \{1, 2\}$) n'améliore pas la valeur $f(x^0) = 1$, et par conséquent l'algorithme CS stagne en ce point.

Un degré de liberté de l'algorithme qu'il est intéressant de mentionner est l'ordre dans lequel les points de P^k sont évalués pour les comparer à la solution courante x^k. Si l'on décide

d'évaluer tous les points, alors l'ordre n'a pas d'intérêt. En revanche, si l'on décide d'évaluer les points de P^k seulement jusqu'à ce que l'on en trouve un meilleur que x^k, alors il vaut mieux que les plus prometteurs soient évalués en premier. Cela s'appelle une stratégie *opportuniste*. Sarrazin-McCann montre et examine dans son mémoire [73] l'intérêt de telles stratégies.

2.1.2 Recherche par motifs généralisée (GPS)

Torczon introduisit en 1997 la recherche par motifs généralisée [82]. C'est un cadre algorithmique qui généralise la recherche par coordonnées, mais aussi d'autres méthodes : la recherche par motifs (*pattern search*) [42], la recherche multidirectionnelle (*multidirectional search*) [31], et une méthode évolutionnaire (*evolutionary operation method*) [22]. Le cadre GPS diffère du cadre CS en trois aspects principaux. Le premier est que les directions de sonde ne sont plus limitées aux seules directions des coordonnées mais peuvent être quelconques et changer d'une itération à l'autre. Le deuxième est l'ajout d'une phase appelée *recherche globale* dont le but est de pouvoir explorer des points qui ne sont pas dans l'ensemble de sonde P^k. Le troisième est que la taille du pas δ^k augmente lors d'un succès, en plus de diminuer lors d'un échec.

Premièrement, l'enrichissement des directions de sonde utilise des *ensembles générateurs positifs* [30].

Définition 2.2 (Ensemble générateur positif) *Soit $\mathbb{D} = \{d^1, d^2, \ldots, d^{|\mathbb{D}|}\}$ un ensemble fini de vecteurs de $\mathbb{R}^n$. $\mathbb{D}$ est un* ensemble générateur positif *de $\mathbb{R}^n$ si tout vecteur $v \in \mathbb{R}^n$ peut s'écrire sous la forme d'un combinaison linéaire positive des vecteurs de $\mathbb{D}$, i.e., sous la forme*

$$v = \sum_{i=1}^{|\mathbb{D}|} \lambda_i d^i, \text{ avec } \lambda_i \geq 0 \text{ pour tout } i \in \{1, 2, \ldots, |\mathbb{D}|\}$$

L'ensemble des directions de coordonnées $\left\{ \pm e_i \ : \ 1 \leq i \leq n \right\}$ utilisé dans l'algorithme CS est un exemple d'ensemble générateur positif. L'intérêt d'un tel ensemble est que si f est une fonction de classe $\mathcal{C}^1$ dont le gradient est non nul en un point x, alors il existe forcément une direction d dans cet ensemble tel que $\nabla f(x)^\top d < 0$, ce qui signifie que d est une direction de descente de f en x. Autrement dit, un ensemble générateur positif est suffisamment représentatif de toutes les directions possibles pour que l'on y trouve au moins une direction de descente. La phase de sonde de l'algorithme GPS consiste à choisir un ensemble générateur positif et à construire des points candidats le long de ses directions. L'ensemble de sonde devient alors

$$P^k = \left\{ x^k + \delta^k d \ : \ d \in \mathbb{D}^k \right\} \tag{2.2}$$

où $\mathbb{D}^k \subseteq \mathbb{D}$ est une sélection de directions à l'itération k parmi celles de $\mathbb{D}$. L'algorithme GPS offre donc la possibilité de choisir des directions de sonde différentes à chaque itération, à condition qu'elles soient incluses dans $\mathbb{D}$ et qu'elles forment un ensemble générateur positif.

Deuxièmement, la nouvelle phase dite de recherche globale nécessite la notion de *treillis*.

Définition 2.3 (Treillis) *soit G une matrice inversible de $\mathbb{R}^{n\times n}$ et Z une matrice de $\mathbb{Z}^{n\times|\mathbb{D}|}$ dont les colonnes forment un ensemble générateur positif de $\mathbb{R}^n$. Posons $D := GZ$. Le* treillis *engendré par D à la solution courante x^k de pas $\delta^k > 0$ est défini par*

$$M^k = \left\{ x^k + \delta^k D y \ : \ y \in \mathbb{N}^{|\mathbb{D}|} \right\}.$$

Comme les colonnes de Z forment un ensemble générateur positif et que G est inversible, les colonnes de D forment aussi un ensemble générateur positif, qui n'est autre de $\mathbb{D}$. L'étape de recherche globale consiste à choisir des points sur ce treillis, c'est-à-dire qu'à chaque itération on évalue f sur un sous-ensemble fini S^k de M^k.

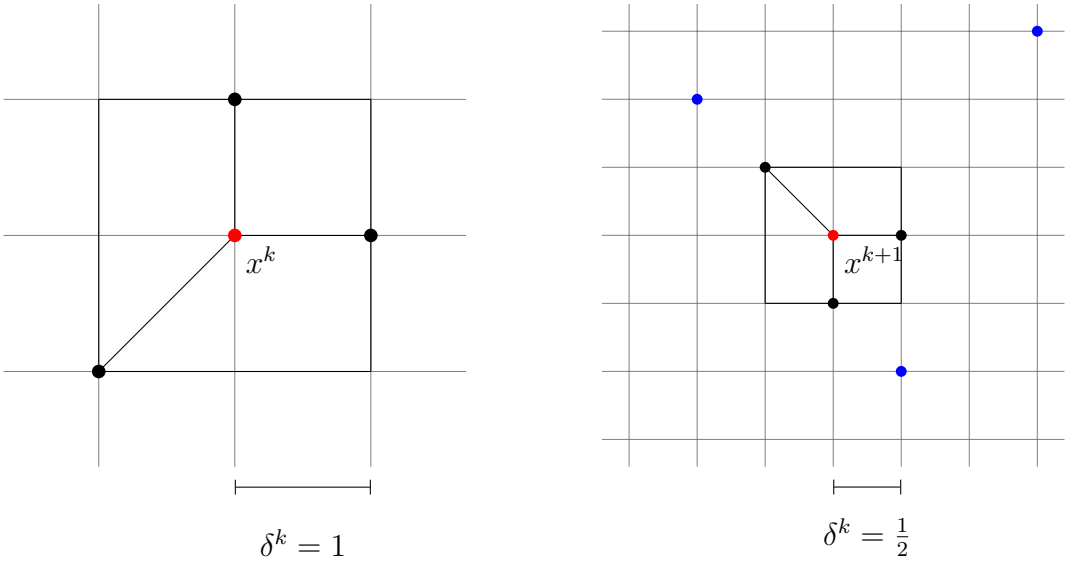


Figure 2.1 Treillis, points de sonde et points de recherche globale dans GPS.

La figure 2.1 illustre la notion de treillis avec $D = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 & 1 \end{bmatrix}$.

Sur l'image de gauche, trois points de sonde sont représentés en noir autour de la solution courante en rouge. À droite, le pas a été diminué de moitié après un échec, les directions de sonde ont changé et trois points de recherche globale ont été représentés en bleu. On voit sur cette figure que les points de sonde reposent tous sur un même cadre.

Troisièmement, on introduit un paramètre d'ajustement du treillis $\tau \in]0,1[$ qui permet de changer la taille du pas δ^k. Si l'itération est un succès, on divise le pas par τ, ce qui l'agrandit ; sinon, on le multiplie par τ, ce qui le diminue. L'idée nouvelle ici est que si l'on trouve un meilleur point dans une certaine direction, alors la solution courante n'est probablement pas proche d'un optimum et on peut par conséquent agrandir le pas pour une exploration plus franche. Dans l'algorithme CS, τ est égal à $1/2$ mais on ne peut que diminuer le pas lors d'un échec, et pas l'agrandir lors d'un succès. L'algorithme 2 formalise le fonctionnement de GPS.

Les résultats de convergence de GPS [14, Chapitre 7] sont similaires à ceux de CS. Si f est de classe $\mathcal{C}^1$ et ses ensembles de niveau sont bornés, alors un point d'accumulation $\hat{x}$ de la suite des itérés ayant conduit à un échec est un point critique de f. Si f est seulement lipschitzienne, alors pour toute direction $d \in \mathbb{D}$ telle que f a été évaluée une infinité de fois dans la suite des itérés ayant conduit à un échec, on a $f^\circ(\hat{x}; d) \geq 0$, où $f^\circ(\hat{x}; d)$ est la dérivée directionnelle généralisée de Clarke [25] de f en $\hat{x}$ dans la direction d

$$f^\circ(\hat{x}; d) = \lim_{y \to \hat{x}} \sup_{t \to 0^+} \frac{f(y + td) - f(y)}{t} \tag{2.3}$$

À nouveau, la non-décroissance est garantie dans un ensemble fini de directions et l'on peut trouver des exemples de problèmes pour lesquels GPS ne converge pas vers un point critique [14, Chapitre 7].

La phase de recherche globale n'a pas d'influence sur les résultats de convergence. Elle a en revanche un intérêt pratique car elle permet d'étendre la recherche au-delà de la phase de sonde et ainsi d'éviter de converger vers des minima locaux. La manière de sélectionner des points candidats lors de la phase de recherche globale est libre tant que les points restent sur le treillis M^k. Si l'utilisateur a une connaissance spécifique du problème à traiter, il peut concevoir une recherche *ad hoc*. Sinon, on peut employer des méthodes d'échantillonnage (aléatoire ou hypercube latin [55]), des heuristiques (recherche à voisinage variable [57], recuit simulé [47], Nelder-Mead [64]) ou bien des méthodes utilisant des substituts qui seront abordées à la section 2.2.

Algorithme 2 : Recherche par motifs généralisée (GPS)

***0. Initialisation* :**

$x^0 \in \mathbb{R}^n$: point initial
$\delta^0 > 0$: pas initial
$D = GZ$: matrice génératrice positive
$\tau \in]0, 1[$: paramètre d'ajustement
$\epsilon_{\text{stop}} > 0$: critère d'arrêt
$k \leftarrow 0$: compteur d'itérations

***1. Recherche globale* :**

Choisir un ensemble de recherche $S^k \subset M^k$
si $\exists\ y \in S^k,\ f(y) < f(x^k)$ **alors**
  $x^{k+1} \leftarrow y$
  $\delta^{k+1} \quad \tau^{-1}\delta^k$
  Aller à 3
sinon
  Aller à 2

***2. Sonde* :**

Choisir un ensemble générateur positif $\mathbb{D}^k \subset \mathbb{D}$
si $\exists\ y \in P^k = \left\{x^k + \delta^k d \ :\ d \in \mathbb{D}^k\right\},\ f(y) < f(x^k)$ **alors**
  $x^{k+1} \quad y$
  $\delta^{k+1} \quad \tau^{-1}\delta^k$
sinon
  $x^{k+1} \quad x^k$
  $\delta^{k+1} \quad \tau\delta^k$

***3. Critère d'arrêt* :**

si $\delta^{k+1} \geq \epsilon_{\text{stop}}$ **alors**
  $k \quad k+1$
  Aller à 1
sinon
  STOP

2.1.3 Recherche directe par treillis adaptatifs (MADS)

Audet et Dennis proposèrent en 2006 la recherche directe par treillis adaptatifs [10]. Ce cadre algorithmique généralise GPS en deux points : une infinité de directions peuvent être choisies à la phase de sonde, contrairement à l'ensemble fini de directions $\mathbb{D}$ dans GPS ; et la méthode s'étend aux problèmes avec contraintes.

Le but d'offrir une infinité de directions de sonde est de pouvoir choisir des points candidats non seulement sur le cadre de sonde comme dans GPS, mais aussi à l'intérieur du cadre, et cela de façon dense. Pour y parvenir tout en gardant des résultats de convergence au moins aussi forts que ceux de GPS, le concept de pas δ^k vu jusqu'à présent laisse place à

deux nouvelles notions : le paramètre de taille du treillis, qui récupère la notation δ^k, et le paramètre de taille du cadre Δ^k. La taille du treillis définit le maillage sur lequel tous les points candidats peuvent se trouver, comme dans GPS. La taille du cadre définit quant à elle les limites de l'ensemble de sonde P^k. Ces paramètres doivent respecter à tout moment $0 < \delta^k \leq \Delta^k$ ainsi que

$$\lim_{t\in T} \delta^k = 0 \iff \lim_{t\in T} \Delta^k = 0 \text{ pour tout sous-ensemble infini d'indices } T. \tag{2.4}$$

La taille du cadre Δ^k est soumise aux mêmes règles de mise à jour que dans GPS, c'est-à-dire qu'elle est multipliée par un paramètre d'ajustement $\tau \in]0, 1[$ en cas d'échec de l'itération, et est divisée par ce même paramètre en cas de succès. La taille du treillis δ^k est libre tant qu'elle respecte (2.4). Toutefois, en pratique on prend $\delta^k = \min\{\Delta^k, (\Delta^k)^2\}$, de sorte que δ^k tend plus vite vers 0 que Δ^k. Cela permet d'avoir un treillis de plus en plus fin à mesure que la taille du cadre de sonde diminue, et par conséquent l'ensemble des directions normalisées engendré au cours de l'algorithme est dense dans la sphère unité.

Comme dans GPS, on se donne une matrice $D = GZ$ dont les colonnes forment un ensemble générateur positif, encore noté $\mathbb{D}$. Tous les points doivent aussi se trouver sur un treillis M^k similaire à celui de la définition 2.3, qui diffère toutefois en ce que son origine n'est pas seulement la solution courante x^k : tous les points où la fonction objectif a été évaluée peuvent servir d'origine. On appelle l'ensemble de ces points la *cache* à l'itération k et on le note V^k.

$$M^k = \left\{ x + \delta^k D y \ : \ x \in V^k, \ y \in \mathbb{N}^{|\mathbb{D}|} \right\} . \tag{2.5}$$

Ainsi M^k n'est plus un treillis centré sur la solution courante mais plutôt la réunion des treillis centrés sur chaque point de la cache.

Pour la phase de sonde, on introduit formellement la notion de *cadre de sonde* à l'itération k, noté F^k, dans lequel tous les points de sonde doivent se trouver

$$F^k := \{ x \in M^k \ : \ ||x - x^k||_\infty \leq \delta^k b \} , \tag{2.6}$$

où $b = \max\{||d'||_\infty : \ d' \in \mathbb{D}\}$. La phase de sonde consiste alors à choisir un ensemble générateur positif $\mathbb{D}_\Delta^k$ tel que l'ensemble de sonde P^k soit inclus dans le cadre de sonde F^k.

La figure 2.2 illustre l'évolution de l'ensemble de sonde. Les directions de sonde sont celles des coordonnées, comme dans CS. À gauche, la taille du cadre est égale à la taille du treillis ($\Delta^k = \delta^k = 1$). Après un échec (au milieu), alors que la taille de cadre est divisée par 2, la

taille de treillis est divisée par 4, ce qui augmente le nombre de points candidats possibles au sein du cadre de sonde. Après un deuxième échec (à droite), la taille de cadre est à nouveau divisée par 2 et la taille de treillis par 4, ce qui augmente encore le nombre de points candidats possibles. On voit qu'en poursuivant cette progression, on peut atteindre un nombre arbitrairement grand de points candidats possibles au sein du cadre de sonde qui, lui, peut être arbitrairement petit.

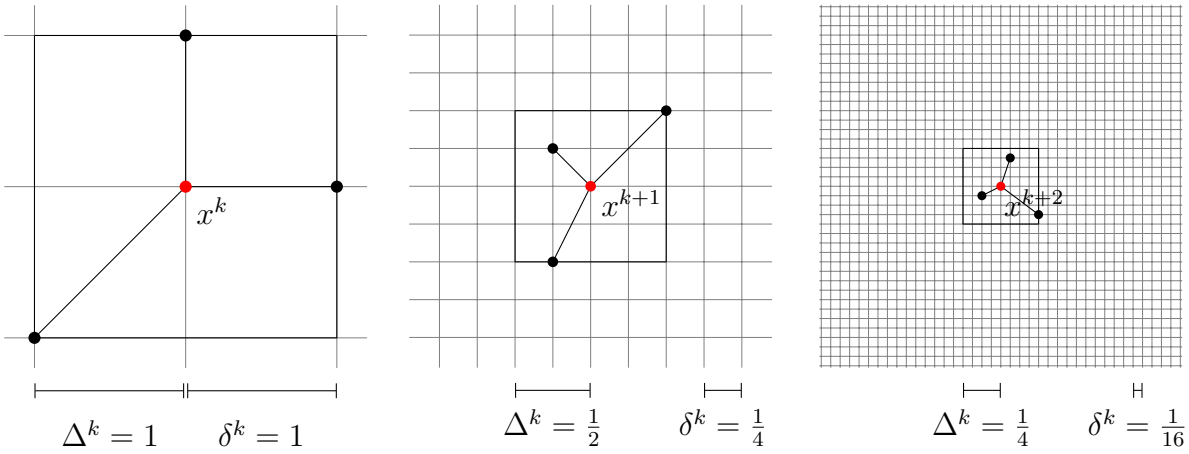


Figure 2.2 Sonde dans MADS.

Dans sa version initiale, MADS gère les contraintes avec la méthode de la barrière extrême (*Extreme Barrier*) (EB). Une méthode plus élaborée fut mise au point par la suite et est décrite dans la section 2.1.4. La méthode EB consiste à remplacer la fonction objectif par

$$f_\Omega(x) = \begin{cases} f(x) & \text{si } x \in \Omega \\ +\infty & \text{sinon} \end{cases} \tag{2.7}$$

où Ω est l'ensemble réalisable du problème (P). Ainsi, les points non admissibles ne sont jamais retenus. L'algorithme 3 formalise le fonctionnement de MADS.

Le résultat de convergence de MADS nécessite les notions de *cône hypertangent*, et de sous-suite, point et direction *raffinants* [14, Chapitre 7].

Définition 2.4 (Cône hypertangent [43]) *Un vecteur $d \in \mathbb{R}^n$ est un* vecteur hypertangent *à l'ensemble $\Omega \subset \mathbb{R}^n$ au point $x \in \Omega$ si et seulement s'il existe un scalaire $\epsilon > 0$ tel que*

$$\forall y \in \Omega \cup B_\epsilon(x),\ \forall t \in]0, \epsilon[,\ \forall w \in B_\epsilon(d),\ y + tw \in \Omega\ .$$

L'ensemble des vecteurs hypertangents à Ω en x est appelé le cône hypertangent *à Ω en x et est noté $T_\Omega^H(x)$.*

Algorithme 3 : Recherche directe par treillis adaptatif (MADS)

0. Initialisation :

$x^0 \in \mathbb{R}^n$: point initial
$\Delta^0 > 0$: cadre de sonde initial
$D = GZ$: matrice génératrice positive
$\tau \in]0, 1[$: paramètre d'ajustement
$\epsilon_{stop} > 0$: critère d'arrêt
$k \leftarrow 0$: compteur d'itérations

1. Mise à jour des paramètres :

$\delta^k \leftarrow \min\{\Delta^k, (\Delta^k)^2\}$

2. Recherche globale :

Choisir un ensemble de recherche $S^k \subset M^k$
si $\exists\ y \in S^k,\ f_\Omega(y) < f_\Omega(x^k)$ **alors**
  $x^{k+1} \leftarrow y$
  $\Delta^{k+1} \leftarrow \tau^{-1}\Delta^k$
  Aller à 4
sinon
  Aller à 3

3. Sonde :

Choisir un ensemble générateur positif $\mathbb{D}_\Delta^k$ tel que $P^k\left\{x^k + \delta^k d\ :\ d \in \mathbb{D}_\Delta^k\right\}$ soit inclus dans le cadre F^k de taille Δ^k
si $\exists\ y \in P^k,\ f_\Omega(y) < f_\Omega(x^k)$ **alors**
  x^{k+1} y
  Δ^{k+1} $\tau^{-1}\Delta^k$
sinon
  x^{k+1} x^k
  Δ^{k+1} $\tau\Delta^k$

4. Critère d'arrêt :

si $\Delta^{k+1} \geq \epsilon_{stop}$ **alors**
  k $k+1$
  Aller à 1
sinon
  STOP

Le cône hypertangent généralise la notion de cône tangent qui est l'ensemble des directions qui pointent vers l'intérieur de l'ensemble Ω en un point donné $x \in \Omega$. Le cône hypertangent a cela de plus qu'il s'applique aussi aux ensembles dont la frontière n'est pas différentiable. Dans un contexte d'optimisation, il représente le cône des solutions réalisables.

Définition 2.5 (Sous-suite, point et direction raffinants) *Une sous-suite convergente d'itérés ayant conduit à un échec $\{x^k\}_{k\in K}$ (pour un certain sous-ensemble d'indices K) est*

appelée une sous-suite raffinante *si et seulement si* $\lim_{k\in K} \delta^k = 0$. *La limite* $\hat{x}$ *de* $\{x^k\}_{k\in K}$ *est appelée un* point raffinant. *Pour une sous-suite raffinante* $\{x^k\}_{t\in K}$ *et le point raffinant correspondant* $\hat{x}$ *donnés, une direction* d *est appelée une* direction raffinante *si et seulement s'il existe un sous-ensemble infini* $U \subseteq K$ *dont les directions de sonde* $d^k \in \mathbb{D}^k_\Delta$ *sont telles que* $x^k + \delta^k d^k \in \Omega$ *et* $\lim_{t\in U} \frac{d^k}{||d^k||} = \frac{d}{||d||}$.

Le résultat de convergence de MADS est contenu dans la proposition 2.6 et le théorème 2.7.

Proposition 2.6 *Soit* $f : \mathbb{R}^n \to \mathbb{R}$ *une fonction dont les ensembles de niveau sont bornées. Alors les suites* $\{\delta^k\}$ *et* $\{\Delta^k\}$ *des paramètres de taille du treillis et de taille du cadre engendrées par l'algorithme MADS sur* f *sont telles que*

$$\liminf_{t\to\infty} \Delta^k = \liminf_{t\to\infty} \delta^k = 0$$

Théorème 2.7 (Convergence de la recherche directe par treillis adaptatifs) *soit* f *une fonction lipschitzienne au voisinage d'un point raffinant* $\hat{x} \in \Omega$ *et* $d \in T^H_\Omega(\hat{x})$ *une direction raffinante pour* $\hat{x}$. *Alors* $f^\circ(\hat{x}; d) \geq 0$.

La proposition 2.6 garantit l'existence d'un point raffinant et, par un argument de compacité de la boule unité, on peut aussi garantir l'existence d'une direction raffinante. Le théorème 2.7 affirme que f n'est pas décroissante en un point raffinant $\hat{x}$ dans une direction raffinante d. Si de plus f est lipschitzienne, alors cela est vrai pour toute direction $d \in T^H_\Omega(\hat{x})$. La supériorité des résultats de convergence de MADS vient de ce qu'ils garantissent la non-décroissance dans toutes les directions du cône des solutions réalisables, alors les résultats de GPS ne garantissent la non-décroissante que dans un ensemble fini de directions.

Dans la première version de MADS, les directions de sonde sont créées de façon aléatoire avec des matrices triangulaires inférieures, d'où le nom *Lower Triangular* MADS (LtMADS) [10]. Deux autres méthodes ont été mises au point : *Orthogonal* MADS (OrthoMADS) [2] qui utilise des matrices de Householder pour créer de façon déterministe - donc reproductible - des directions orthogonales entre elles ce qui permet une meilleure couverture de l'ensemble des directions possibles ; et QrMADS [85], conçue pour être plus efficace que OrthoMADS en grande dimension, qui utilise la décomposition QR pour créer de façon aléatoire des directions quasi-orthogonales.

2.1.4 Gestion des contraintes

Dans le cadre de ce projet, on distingue trois types de contraintes selon la taxonomie des contraintes en DFO et BBO établie par Le Digabel et Wild [52] : les contraintes relaxables, non-relaxables et cachées.

Comme énoncé dans l'introduction, $\mathcal{X}$ est l'ensemble des points qui respectent les contraintes dites *non-relaxables*, c'est-à-dire les contraintes qui ne doivent jamais être violées, soit parce que la fonction objectif n'est pas définie hors de $\mathcal{X}$, soit parce que les points hors de cet ensemble n'ont pas d'intérêt ou pas de sens pour le problème. Par exemple, une longueur ou une masse ne peut pas être négative, et une probabilité ne peut pas être supérieure à un. Inclus dans $\mathcal{X}$ se trouve l'ensemble réalisable Ω, qui est défini par les contraintes c_j, $j \in \{1, 2, \ldots, m\}$, dites *relaxables*. Ces contraintes sont imposées dans le problème mais il peut être intéressant d'explorer des points dans $\mathcal{X}\backslash\Omega$. Par exemple si une contrainte relaxable représente un budget à ne pas dépasser, une solution qui ne satisfait pas cette contrainte peut quand même être exploitable pour un décideur. Enfin, parmi les contraintes non-relaxables, on distingue les contraintes dites *cachées*. Ce sont des contraintes qui ne sont pas connues *a priori* mais qui excluent une partie des solutions réalisables au sens des autres contraintes. Cela arrive typiquement lorsque la fonction objectif est le résultat d'un code informatique, et qu'exécuter ce code en un certain point $x \in \mathcal{X}$ conduit à une erreur et aucun résultat n'est obtenu. On dit alors que le point x viole une contrainte cachée.

Dans sa version originale [10], MADS gère les contraintes avec la méthode la barrière extrême (EB) dans laquelle on remplace la fonction objectif f par f_Ω définie à l'équation (2.7). C'est une façon simple de rejeter les points non admissibles, mais qui ne fait pas de distinction entre les ensembles $\mathcal{X}$ et Ω, alors que comme on l'a expliqué plus haut, il peut être opportun d'explorer $\mathcal{X}\backslash\Omega$. On introduit maintenant la *fonction de violation des contraintes* [11]

$$h(x) := \begin{cases} \sum_{j=1}^{m} \left(\max\{0, \hat{c}_j(x)\}\right)^2 & si\ x \in \mathcal{X} \\ +\infty & sinon \end{cases} \tag{2.8}$$

La fonction h est exploitée dans la méthode de la barrière progressive (*Progressive Barrier*) (PB) [11] inspirée des méthodes de filtre [34, 35] dans lesquelles on effectue une optimisation bi-objectif sur f et h en privilégiant les points réalisables par rapport aux points ayant une petite valeur de f mais non réalisables.

Dans la méthode PB, on introduit un seuil positif $h^k_{\max}$ qui est initialisé à $+\infty$ et ne peut que décroître au cours des itérations. Tout point dont la valeur par la fonction h excède $h^k_{\max}$

à l'itération k est rejeté. Les points retenus sont alors comparés deux à deux en utilisant la notion de dominance [14, Chapitre 12].

Définition 2.8 (Points dominés) *Un point réalisable $x \in \Omega$* domine *un autre point réalisable $y \in \Omega$ si $f(x) < f(y)$. On note $x \prec_f y$.*

Un point non réalisable $x \in \mathcal{X}\backslash\Omega$ domine *un autre point non réalisable $y \in \mathcal{X}\backslash\Omega$ si $f(x) \leq f(y)$ et $h(x) \leq h(y)$, avec au moins une inégalité stricte. On note $x \prec_h y$.*

Un point x dans un ensemble donné $S \in \mathcal{X}$ est non dominé *s'il n'est dominé par aucun point de S.*

La figure 2.3 illustre la notion de dominance. Les points noirs représentent les points non dominés. La meilleure solution réalisable, c'est-à-dire celle qui a la plus petite valeur de f, se trouve sur l'axe $h = 0$ et est notée x^r. La meilleure solution non réalisable, c'est-à-dire la solution non dominée ayant la plus petite valeur de f tout en ayant une valeur de h inférieure à $h^k_{\max}$, est notée x^{nr}. Il y a, en plus de x^{nr}, neuf points non réalisables dont trois sont non dominés et cinq sont dominés. La zone grise représente l'ensemble des points non réalisables dominés et des points non réalisables dont la valeur de h excède $h^k_{\max}$, comme le point le plus à droite.

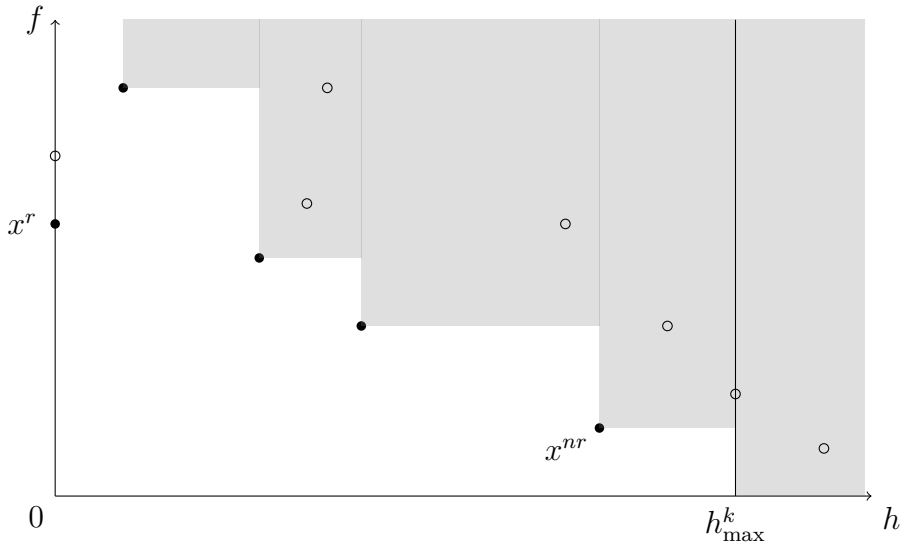


Figure 2.3 Dominance dans la méthode de la barrière progressive.

Lorsqu'on utilise la méthode PB, il n'y a pas une seule solution courante mais deux : la solution réalisable x^r et la solution non réalisable x^{nr}. Deux sondes locales sont donc effectuées. Une itération peu conduire à trois situations différentes :

— on trouve un point qui domine une solution courante, *i.e.*, soit un point réalisable $y \in \Omega$ tel que que $y \prec_f x^r$ soit un point non réalisable $y \in \mathcal{X}\backslash\Omega$ tel que $y \prec_h x^{nr}$, auquel cas $h^{k+1}_{\max} \leftarrow h(x^{nr})$
— on trouve un point qui améliore la valeur de h, *i.e.*, un point non réalisable $y \in \mathcal{X}\backslash\Omega$ tel que $0 < h(y) < h(x^{nr})$, auquel cas $h^{k+1}_{\max} \quad \max\{h(v) \ : \ h(v) < h(x^{nr}),\ v \in V^k\}$ où V^k est la cache à l'itération k
— aucune des deux situations précédentes ne se produit, auquel cas $h^{k+1}_{\max} \leftarrow h(x^{nr})$

L'algorithme MADS avec PB renvoie en sortie les deux solutions courantes x^r et x^{nr} ce qui permet d'estimer l'intérêt qu'il peut y avoir à relaxer certaines contraintes.

Une approche mixte entre la barrière extrême et la barrière progressive consiste à utiliser, pour une contrainte donnée c_j, $j \in \{1, 2, \ldots, m\}$, la barrière progressive jusqu'à obtenir une solution satisfaisant c_j, puis continuer avec la barrière extrême. On appelle cette méthode la barrière progressive-extrême (PEB) [13].

2.2 Méthodes utilisant des substituts

En optimisation de boîtes noires, on peut utiliser des fonctions *substituts* qui, comme leur nom l'indique, remplacent l'objectif ou les contraintes dans la recherche d'un optimum, que l'on appelle alors les *vraies* fonctions du problème. Un substitut est donc censé imiter le comportement de la vraie fonction tout en étant moins coûteux en temps de calcul. Ainsi, l'exploitation de ce substitut permet de guider l'optimisation du vrai problème. Formellement, on note $\tilde{f}$ un substitut de l'objectif f, et $\tilde{c}_j$ un substitut de la contrainte c_j, $j \in \{1, 2, \ldots, m\}$. Une façon simple de formuler le problème substitut est de remplacer l'objectif et les contraintes par leur substitut dans (P) :

$$\begin{array}{ll} \min\limits_{x \in \mathcal{X}} & \tilde{f}(x) \\ \text{s.c.} & \tilde{c}_j(x) \leq 0, \quad \forall j \in \{1, 2, \ldots, m\} \ . \end{array} \qquad (\tilde{P})$$

On distingue deux types de substituts : les *statiques* et les *dynamiques* [14, Chapitre 13]. Un substitut statique est une version simplifiée de la vraie fonction. Cela peut être suivant le contexte : un maillage plus grossier dans la méthode des éléments finis, un critère d'arrêt plus souple dans une méthode numérique, ou un modèle physique plus simple (négliger les frottements, remplacer les équations de Navier-Stokes par celles d'Euler en mécanique des fluides). Un tel substitut est fixé dès le départ et n'évolue pas au cours de l'optimisation, d'où l'adjectif *statique*. Un substitut dynamique est quant à lui un modèle approché de la vraie fonction qui est construit à partir des valeurs déjà connues de celle-ci, et qui est mis

à jour au fur et à mesure de l'optimisation. On appelle aussi un substitut dynamique un *modèle* puisqu'il a vocation à imiter la forme de la vraie fonction, ce qui n'est pas nécessairement le cas d'un substitut statique. Les substituts dynamiques couramment utilisés sont des modèles d'interpolation ou de régression comme les modèles polynomiaux (*Polynomial Response Surface*) (PRS) [3,63] et en particulier quadratiques [26,28,29], les fonctions à base radiale (*Radial Basis Function*) (RBF) [19, 48, 68–70, 80, 91, 92], les machines à vecteurs de support (*Support Vector Machines*) (SVM) [86], le lissage par noyaux (*Kernel Smoothing*) (KS) [3], ou les processus gaussiens (*Gaussian Process*) (GP) [20, 66, 67, 72]. Pour une revue de littérature sur les méthodes utilisant des substituts, voir [15, 89] et [50, Section 2.2].

2.2.1 Utilisation de substituts

Les substituts sont principalement utilisés à deux fins dans un algorithme d'optimisation de boîtes noires : trouver de nouveaux points candidats et ordonner des points candidats existants avant de les évaluer sur le vrai problème. Par exemple, dans un algorithme de recherche directe, un substitut peut être exploité à la phase de recherche globale afin de trouver des points candidats. Une façon simple de procéder est d'appliquer un algorithme d'optimisation de son choix sur le problème substitut puis de prendre la meilleure solution comme nouveau point candidat. Quant à la phase de sonde, les points candidats sont prescrits par l'algorithme mais on a le choix de l'ordre dans lequel ils sont évalués sur le vrai problème. Si l'on procède de façon opportuniste, *i.e.*, en n'évaluant les points candidats que jusqu'à trouver une amélioration, alors cet ordre est important puisqu'on a intérêt à trouver une amélioration le plus vite possible. On peut donc évaluer les points candidats sur le problème substitut, puis les ordonner en fonction de la valeur obtenue. Ainsi, les points les plus prometteurs sont évalués les premiers.

Si des modèles sont utilisés, les mettre à jour lorsque de nouvelles valeurs du vrai problème sont connues permet de les rendre plus fidèles aux vraies fonctions. On a donc intérêt à ne pas concentrer les recherches uniquement dans les zones prometteuses du problème substitut, car alors on n'explore pas assez l'espace et par conséquent les modèles, qui ne reposent que sur les valeurs connues du vrai problème, s'en trouveront moins fiables.

Les fondements des méthodes d'optimisation utilisant des substituts ou modèles, qui reprennent ce qui vient d'être dit en ce début de section, furent posés par Booker et al. avec le *Surrogate Management Framework* [21] formalisé dans l'algorithme 4. Dans l'algorithme MADS, un certains nombre de travaux ont été menés pour y intégrer des substituts : des modèles quadratiques [26], des modèles gaussiens par arbres (*treed gaussian processes*) [38], des modèles LOWESS [78], des modèles hybrides entre statique et dynamique [9], ou encore

des ensembles de modèles [16]. Ces derniers seront vus en détails dans la sous-section 2.2.3.

Algorithme 4 : *Surrogate management framework*

1. ***Exploration*** **:**
   Utiliser le problème substitut pour créer une liste $\mathcal{L}$ de points candidats
   Évaluer le vrai problème aux points de $\mathcal{L}$ de façon opportuniste
   si *une nouvelle solution est trouvée* **alors**
   | Aller à 3
   sinon
   └ Aller à 2
2. ***Classement*** **:**
   Utiliser l'algorithme d'optimisation pour créer une liste $\mathcal{L}$ de points candidats
   Utiliser le problème substitut pour classer les points de $\mathcal{L}$
   Évaluer le vrai problème aux points de $\mathcal{L}$ de façon opportuniste
3. ***Mise à jour des paramètres*** **:**
   Mettre à jour les paramètres algorithmiques
   Vérifier les critères d'arrêt ou aller à 4
4. ***Mise à jour du modèle (optionnel)*** **:**
   Mettre à jour le modèle en utilisant les nouvelles valeurs du vrai problème obtenues en 1 et 2

2.2.2 Optimisation bayésienne

Lorsque des modèles sont utilisés, on peut espérer qu'ils soient utiles et précis dans les zones où les vraies fonctions ont été échantillonnées. En revanche, plus on s'éloigne de ces zones, moins ils sont fiables. Quantifier l'incertitude que l'on a sur la valeur du vrai problème en un point donné de l'espace permet de se réserver la possibilité d'explorer des zones où l'incertitude est grande et ainsi de ne pas consacrer tout le budget d'évaluations dans des zones qui peut-être s'avèreront peu intéressantes. C'est le compromis entre *exploration* et *intensification*.

En optimisation bayésienne, l'objectif est interprété comme un processus stochastique. On suppose une distribution *a priori* $P[f]$, où f désigne toujours la fonction objectif, puis, avec l'ensemble $\mathbb{X}$ des points échantillonnés sur le vrai problème et un modèle de vraisemblance $P[\mathbb{X} \mid f]$, on construit la distribution *a posteriori* $P[f \mid \mathbb{X}]$ grâce à la règle de Bayes

$$P[f \mid \mathbb{X}] = \frac{P[\mathbb{X} \mid f]P[f]}{P[\mathbb{X}]} \ .$$

En des termes plus concrets, choisir une distribution *a priori* consiste à choisir un modèle stochastique et ses paramètres, et construire la distribution *a posteriori* consiste à ajuster ce modèle aux points d'échantillonnage évalués sur la vraie fonction. La loi *a posteriori* caracté-

rise complètement le modèle stochastiques, dont sa moyenne et sa variance. Cela permet de disposer en un point donné $x \in \mathcal{X}$ d'une prédiction $\mu(x)$ mais aussi d'une incertitude $\sigma(x)$. La double information ainsi disponible est exploitée pour choisir un nouveau point candidat. On utilise pour cela une *fonction d'acquisition* qui est censée faire naturellement le compromis entre exploration et intensification.

Les modèles stochastiques utilisés concrètement sont par exemple des modèles linéaires généralisés [65], des processus gaussiens [67], ou des arbres dynamiques [77] (librairie DynaTree [39]). Pour beaucoup de modèles stochastiques, et en particulier pour les processus gaussiens [67] très utilisés en pratique, le modèle de l'objectif suit en un point donné x une loi normale $\tilde{f}(x) \sim \mathcal{N}\{\mu(x), \sigma^2(x)\}$. Les trois fonctions d'acquisition classiques décrites ci-après exploitent cette normalité.

La probabilité d'amélioration (*Probability of Improvement*) (PI) [44] est la probabilité que l'objectif en un point donné soit meilleur qu'une cible prescrite τ, qui peut être par exemple la meilleure valeur connue :

$$\begin{aligned} PI(x) &= \mathbb{P}[\tau - \tilde{f}(x) > 0] \\ &= \Phi\left(\frac{\tau - \mu(x)}{\sigma(x)}\right) \end{aligned} \tag{2.9}$$

où Φ est la fonction de répartition de la loi normale centrée réduite.

L'amélioration espérée (*Expected Improvement*) (EI) [58] ne prend pas seulement en compte la probabilité d'amélioration, mais aussi l'importance de l'amélioration par rapport à une cible τ :

$$\begin{aligned} EI(x) &= \mathbb{E}[\max(\tau - \tilde{f}(x), 0)] \\ &= (\tau - \mu(x))\Phi\left(\frac{\tau - \mu(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\tau - \mu(x)}{\sigma(x)}\right) \end{aligned} \tag{2.10}$$

où ϕ est la fonction de densité de la loi normale centrée réduite. Cette mesure est toujours positive.

La limite supérieure de confiance (*Upper Confidence Bound*) (UCB) [76] ne tient compte que de l'estimation la plus optimiste :

$$UCB(x) = \mu(x) - \kappa\sigma(x) \tag{2.11}$$

où κ est un paramètre de l'algorithme.

La figure 2.4 montre un exemple de régression sur processus gaussien - aussi appelée *krigeage* - ainsi que le graphe de l'amélioration espérée qui en découle. La courbe pointillée rouge représente l'objectif $f : x \mapsto x \sin x$; les points rouges les points d'échantillonnage ; la courbe bleue la prédiction moyenne $\mu : x \mapsto \mu(x)$; et la zone bleue l'intervalle de confiance à 95%, donné en un point x par $[\mu(x)+1.96\sigma(x),\ \mu(x)-1.96\sigma(x)]$. En bas, la courbe verte représente - dans une échelle différente - l'amélioration espérée EI. Le point candidat résultant de cet exemple est celui qui maximise EI, indiqué par la ligne pointillée verticale.

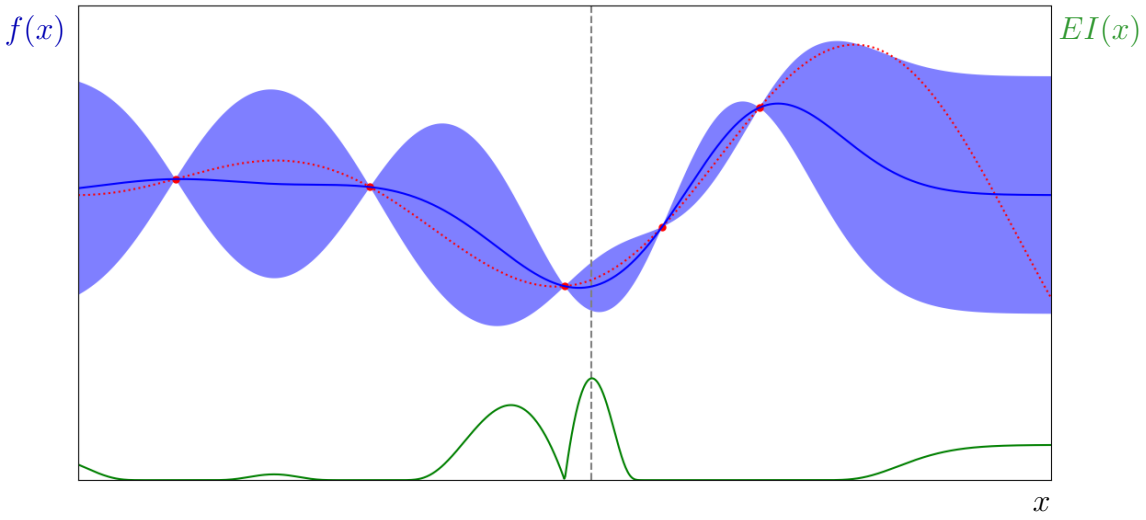


Figure 2.4 Krigeage et amélioration espérée (EI) sur la fonction $f : x \mapsto x \sin x$.

Maximiser EI est une méthode introduite par Jones et al. [45] qui est simple à comprendre et efficace. Des méthodes plus élaborées ont été développées depuis. Talgorn et al. [79] proposent différentes formulations du problème substitut utilisant plusieurs fonctions d'acquisition à la fois. Pour une revue de littérature en optimisation bayésienne, voir [40, 74].

2.2.3 Ensembles de substituts

Dans un contexte d'optimisation, il n'y a pas un seul type de modèle qui domine les autres [3, 37]. Même pour un problème donné, le choix d'un meilleur modèle peut varier en fonction de l'échantillonnage initial [37]. C'est pourquoi des méthodes utilisant plusieurs modèles en même temps se montrent efficaces [3, 16, 23, 37, 63, 87, 93]. L'idée directrice est de construire, à partir de plusieurs modèles de différents types, un modèle agrégé de la forme

$$\hat{f}(x) = \sum_{p=1}^{s} w^p \tilde{f}^p(x) \tag{2.12}$$

avec $s \geq 1$; où $\tilde{f}^1, \tilde{f}^1, \ldots, \tilde{f}^s$ sont s modèles de l'objectif f construits à partir des points d'échantillonnage ; et $w^1, w^2, \ldots, w^s$ sont des poids positifs tels que $\sum_{p=1}^{s} w^p = 1$. On se restreindra dans ce mémoire au cas où les w^p ne dépendent pas de x (voir [23] pour un exemple de cas où $w^p = w^p(x)$). La difficulté réside essentiellement dans l'attribution de la valeur de ces poids. Ils doivent refléter la qualité des modèles de telle sorte que les meilleurs aient plus d'influence. Reste à définir ce qu'est un bon modèle. Il faut pour cela disposer d'une mesure de la qualité, ou, de façon équivalente, de l'erreur que commet un modèle sur l'ensemble des points d'échantillonnage. On note $\mathcal{E}^p$ la mesure de l'erreur du modèle $\tilde{f}^p$, ainsi que $\mathcal{E}^{\text{ag}}$ l'erreur du modèle agrégé.

Une façon de produire une telle mesure est de calculer une métrique statistique par validation croisée. Les plus utilisées sont l'écart quadratique moyen (*Root Mean Square Error*) (RSME) et la somme prédite des carrés résiduels (*Predicted Residual Sum of Squares*) (PRESS). De cette façon, on tient compte de l'écart entre les valeurs du modèle et celles de l'objectif. Dans [16], la métrique d'erreur n'est pas une quantité statistique mais une mesure de la capacité d'un modèle à classer les points en fonction de leur valeur dans le même ordre que le vrai objectif. Cela est motivé par l'idée que ce que l'on attend d'un modèle dans un contexte d'optimisation n'est pas d'approcher au mieux les valeurs de la fonction objectif mais plutôt de pouvoir comparer les points candidats entre eux dans le but de discriminer au mieux les plus prometteurs.

La métrique, censée refléter la qualité d'un modèle, sert ensuite à calculer les poids. Deux approches sont discernables dans la littérature. L'une consiste à calculer la métrique pour chaque modèle puis à attribuer les poids en conséquence [16, 37, 63]. Dans [37], trois façons d'attribuer le poids en fonction de la métrique sont utilisées :

$$\begin{aligned} w^p &\propto \mathcal{E}^{\text{tot}} - \mathcal{E}^p \\ w^p &\propto \mathbb{1}_{\mathcal{E}^p = \mathcal{E}^{\min}} \\ w^p &\propto (\mathcal{E}^p + \alpha \mathcal{E}^{\text{moy}})^\beta \end{aligned} \tag{2.13}$$

où $\mathcal{E}^{\text{tot}}$ est la somme des erreurs de tous les modèles, $\mathcal{E}^{\min}$ est l'erreur minimale, $\mathcal{E}^{\text{moy}}$ est l'erreur moyenne, et $\alpha < 1$ et $\beta < 0$ sont des paramètres ajustables.

L'autre approche consiste à calculer les poids de façon à minimiser la métrique calculée sur le modèle agrégé [3, 87], ce qui donne lieu à un sous-problème d'optimisation de la forme

$$\begin{aligned} \min_{w \in \mathbb{R}^s} \quad & \mathcal{E}^{\mathrm{ag}} \\ \text{s.c.} \quad & \sum_{p=1}^{s} w^p = 1 \\ & w \geq 0 \end{aligned} \tag{2.14}$$

où l'on a noté le vecteur des poids $w = [w^1, \ldots, w^s]^\top$. Une approche hybride est adoptée dans [93]. Les poids sont attribués selon la formule $w^p \propto (\mathcal{E}^p + \alpha \mathcal{E}^{\mathrm{moy}})^\beta$, mais les valeurs α et β sont calculées de façon à minimiser $\mathcal{E}^{\mathrm{ag}}$.

CHAPITRE 3 DÉMARCHE DU TRAVAIL

La contribution apportée dans ce mémoire concerne les ensembles de modèles. Motivés par le constat que l'optimisation bayésienne d'une part, et l'emploi d'ensembles de modèles d'autre part produisaient des méthodes efficaces, nous avons pensé que la combinaison des deux philosophies serait féconde.

Utiliser un ensemble de modèles dans le cadre de l'équation (2.12) produit un modèle agrégé $\hat{f}$ qui est déterministe, ce qui est incompatible avec l'optimisation bayésienne. Toutefois, si plusieurs modèles sont utiles pour décrire une même fonction, c'est précisément parce que leurs prédictions ne sont pas identiques. On peut donc s'attendre à ce qu'il y ait des zones de l'espace de recherche dans lesquelles les modèles ne s'accordent pas, créant ainsi une forme d'incertitude qu'il importe de mesurer.

L'idée que l'on puisse identifier des zones où l'incertitude est grande en fonction de l'hétérogénéité des modèles a déjà été abordée dans la littérature. Dans [37], l'incertitude en un point x est donnée par l'écart type entre les valeurs des modèles et est utilisée une fois l'optimisation finie pour vérifier que les zones où l'incertitude est importante sont aussi celles où l'erreur du modèle agrégé est grande. Dans [60], deux substituts sont disponibles : un *haute-fidélité* (précis mais coûteux) et un *basse-fidélité* (peu précis mais peu coûteux), et un coefficient de corrélation entre modèles est utilisé pour décider localement quel substitut évaluer. Dans [81], l'importance de la corrélation entre modèles de fidélité variable est analysée plus en détails et en particulier comment elle peut guider le choix du modèle à évaluer. Dans [71], on dispose de plusieurs modèles gaussiens (*co-kriging*) de niveau de fidélité variable et la probabilité d'amélioration (PI) est modifiée pour prendre en compte, entres autres, la corrélation entre les modèles. Tous ces travaux font appel à des substituts ou modèles spécifiques et ne s'inscrivent pas dans le cadre des ensembles de modèles tel que décrit dans la section 2.2.3.

Notre approche, quand à elle, concerne les ensembles de modèles et consiste à fournir une mesure de la divergence entre les modèles et de l'interpréter comme une incertitude. Cela donne un modèle agrégé qui, en tout point x, fournit non seulement une prédiction $\hat{f}(x)$ issue de l'équation (2.12), mais aussi une incertitude $\hat{\sigma}(x)$ qui n'a pas la même expression si la fonction est l'objectif ou une contrainte. Cela constitue une nouveauté en ce que cette approche produit une expression de l'incertitude pour une sélection arbitraire de modèles, propre au rôle de la fonction dans le problème (objectif ou contrainte), et qui permet de bénéficier des outils de l'optimisation bayésienne.

CHAPITRE 4 ARTICLE 1: QUANTIFYING UNCERTAINTY WITH ENSEMBLES OF SURROGATES FOR BLACKBOX OPTIMIZATION

Charles Audet · Sébastien Le Digabel · Renaud Saltet

Submitted to *Computational Optimization and Applications*

Abstract: This work is in the context of blackbox optimization where the functions defining the problem are expensive to evaluate and where no derivatives are available. A tried and tested technique is to build surrogates of the objective and the constraints in order to conduct the optimization at a cheaper computational cost. This work proposes different uncertainty measures when using ensembles of surrogates. The resulting combination of an ensemble of surrogates with our measures behaves as a stochastic model and allows the use of efficient Bayesian optimization tools. The method is incorporated in the search step of the mesh adaptive direct search (MADS) algorithm to improve the exploration of the search space. Computational experiments are conducted on seven analytical problems, two multidisciplinary optimization problems and two simulation problems. The results show that the proposed approach solves expensive simulation-based problems at a greater precision and with a lower computational effort than stochastic models.

Keywords: Blackbox optimization, Derivative-free optimization, Ensembles of surrogates, Mesh adaptive direct search, Bayesian optimization

4.1 Introduction

This work considers the constrained optimization problem

$$\begin{array}{ll} \min\limits_{x\in\mathcal{X}} & f(x) \\ \text{s.t.} & x \in \Omega \end{array} \qquad (P)$$

where $\mathcal{X}$ is a subset of $\mathbb{R}^n$; Ω denotes the feasible set $\Big\{x \in \mathcal{X} \mid c_j(x) \leq 0,\ 1 \leq j \leq m\Big\}$, where the functions $c_j : \mathbb{R}^n \to \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ are the constraint functions of the problem; and $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is the objective function. The set $\mathcal{X}$ contains the points that

satisfy *unrelaxable* constraints [51]: every point explored during the optimization process must lie in $\mathcal{X}$ either because f is not defined elsewhere or because a point outside $\mathcal{X}$ has no meaning in the original problem, e.g., a negative length or a probability greater than one. The set $\mathcal{X}$ typically represents bound constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ where ℓ and u are vectors of $\{\mathbb{R} \cup \{-\infty\}\}^n$ and $\{\mathbb{R} \cup \{+\infty\}\}^n$, respectively. The functions c_j, $j \in \{1, 2, \ldots, m\}$, denote *relaxable* constraints, which means that they can be violated during the optimization process, however, the final solution must satisfy these constraints.

In *blackbox optimization* (BBO), no information is available on the functions f and c_j, $j \in \{1, 2, \ldots, m\}$, beyond the mere values they produce, hence the *blackbox* designation. In particular, no derivatives can be used either because they are especially hard to estimate or because they do not exist. Designing algorithms that do not use derivatives is referred to as *Derivative-Free Optimization* (DFO). This context typically occurs when the functions are the results of numerical simulations. Consequently, a blackbox is assumed to be costly, i.e., one evaluation might take seconds [24], minutes [7, 62], hours [18, 83] or even days [54]. BBO consists in designing algorithms capable of finding the best possible solution to such a problem with a given budget of function evaluations. For a better understanding of the theoretical importance of the existence of derivatives in DFO, see [15]. Two reference books are available in DFO and BBO [14, 27] as well as an extended review [50].

BBO algorithms can be roughly divided in two categories: *direct-search* methods and methods using *surrogates*. Direct-search algorithms only use comparisons between points and no other information like an approximation of derivatives. This philosophy has led to fruitful algorithmic frameworks such as the state-of-the-art algorithms *generalized pattern search* (GPS) [82] and *mesh-adaptive direct search* (MADS) [10], the latter is described in this article. Both frameworks lie on the search-poll paradigm [5]: the *search* step offers flexibility for the user to implement any method they see fit for the problem to optimize, while the *poll* step imposes more rigid procedures in order to guaranty convergence and explore further the surroundings of the best known solution. The second category resorts to *surrogates*, i.e., functions that are expected to mimic the behaviour of the objective and the constraints while being significantly cheaper to evaluated. A surrogate can be a simplified and static version of the blackbox that do not evolve over the optimization, or a dynamic surrogate that is based on regression or interpolation on the previously evaluated points. A dynamic surrogate is also called a model. One can then expect that minimizing a surrogate of the objective while satisfying surrogates of the constraints will lead to a promising new candidate point for the true problem. The combination of direct-search and model-based methods has proven fruitful. Surrogates can be incorporated either in a subproblem embedded in the search step to find candidates points, or as means of ranking candidate points of the poll

step when opportunistic strategies are used.

Among the model-based methods, ensembles of models and stochastic models are two efficient techniques. Ensembles of models consist in giving each model a weight that is supposed to reflect its quality, and the combination of all weighted models yields an *aggregate model* that can be used as a standard surrogate. Stochastic models not only produce a prediction at a given point but also a measure of the uncertainty on this prediction, which is fit for Bayesian optimization. Ensembles of models and stochastic models have both proven efficient but remain fundamentally separated. Using several deterministic surrogates naturally produces a deterministic aggregate model which is incompatible with Bayesian optimization. If some of the models used are stochastic though, the provided uncertainty can be exploited by all the models as in [88]. However, typical stochastic surrogates as Gaussian processes (GPs) become particularly costly to train as the training set grows. The proposed approach is to identify areas where the predictions of the models differ from each other in order to derive some form of uncertainty. The idea of using the correlation between several models to guide the optimization has already been tackled in the literature. In [37], the deviation between the predicted values of several models is used *a posteriori* to check the overall quality of the aggregate model. In [60], two surrogates of the objective are available: a low-fidelity one that is cheap to compute but not accurate and a high-fidelity one that is in contrast more expensive and more reliable. Then RBF models of the two surrogates are computed and the correlation between them is used to choose which surrogate, high or low-fidelity, to evaluate next. In [81], the correlation between variable fidelity co-kriging models is exploited to determine which model to evaluate. In [71], the correlation between variable fidelity multi-level generalized co-kriging models is incorporated in an extended probability of improvement.

The contribution of this work is an extension to ensembles of models when used in the form of aggregate models. For a point x of the search space $\mathcal{X}$, the extended aggregate models produce not only a prediction $\hat{f}(x)$, but also an uncertainty $\hat{\sigma}(x)$, therefore imitating a stochastic model. The resulting surrogate is then exploited in the search step of MADS in subproblems inspired by Bayesian optimization. The proposed approach has been tested on seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. It has been compared to other versions of MADS as well as two other BBO solvers. Results show that the proposed extended aggregate models manage to find solutions of most of the difficult real-world problems at a greater precision than the other algorithms, and with less computational effort than the competing stochastic models.

The manuscript is structured as follows. Section 4.2 introduces ensemble of surrogates and the Bayesian optimization framework as well as a high-level description of the MADS algorithm.

Section 4.3 describes the quantification of uncertainty when using ensembles of models and our incorporation of the resulting extended aggregtae models into the MADS algorithm. Section 4.4 shows the computational results on the set of problems. A concluding discussion is proposed in Section 4.5.

4.2 Background

This section describes the use of surrogates in BBO, with a special focus on stochastic surrogates and ensembles of surrogates, as well as the MADS algorithm.

4.2.1 Surrogates in BBO

A common approach in BBO uses *surrogates* of the objective and the constraints in order to guide the optimization. A surrogate shares similarities with the true functions of the problem while being significantly cheaper. Two types of surrogates can be distinguished: *static surrogates* and *dynamic models*. A static surrogate is a simplified version of the blackbox that can be obtained for example through a simplified physics model, a coarser mesh in a finite elements simulation or a looser stopping criterion in a numerical method. Such a surrogate is fixed and does not evolve over the optimization, hence the *static* designation. On the other hand, a dynamic model is an interpolation or regression model that approaches the true functions by fitting previously evaluated sample points. Since it attempts to approximate the true function, the *model* designation is more appropriate than surrogate in this case. Common models used in BBO are polynomial response surfaces (PRS) [3, 63] and especially quadratic models [26, 28, 29], radial basis functions (RBF) [19, 48, 68–70, 80, 91, 92], support vector machines (SVM) [86], kernel smoothing (KS) [3], and Gaussian processes (GPs) [20, 66, 67, 72].

Surrogates are basically used for two purposes in BBO: finding new candidate points and ranking existing candidate points before evaluation by the true problem. The surrogate management framework [21] establishes the interplay between the surrogate evaluations and the true evaluations and is described in Algorithm 5 as in [14, Chapter 13].

For extended reviews on model-based or model-assisted optimization, see [15, 90] and [50, Section 2.2].

Algorithme 5 : Surrogate management framework.

1. ***Exploration using the surrogate***
   Use the surrogate problem to generate a list $\mathcal{L}$ of candidate points
   Evaluate the true functions at points in $\mathcal{L}$ in an opportunistic way
   If a new incumbent solution is found, go to 3; otherwise go to 2
2. ***Ranking using the surrogate***
   Use the optimization algorithm to generate a list $\mathcal{L}$ of candidate points
   Use the surrogate functions to order the points in $\mathcal{L}$
   Evaluate the true functions at points in $\mathcal{L}$ in an opportunistic way
3. ***Parameters update***
   Update algorithmic parameters
   Check stopping criteria or go to 4
4. ***Model update (optional)***
   Update the model by using the new values of the true functions obtained in 1 and 2

Ensemble of models

In an optimization context, there is not one type of models that dominates the others [3, 37]. Even for a given problem, the best performance can be obtained with different models depending on the initial sampling [37]. A tempting strategy is to resort to several models simultaneously. This idea has proven efficient in several works [3,16,23,37,63,87,93] in which an ensemble of models is used to build an aggregate model defined by

$$\hat{f}(x) = \sum_{p=1}^{s} w^p \tilde{f}^p(x) \tag{4.1}$$

where $s \geq 1$ is the number of models; $\tilde{f}^1, \tilde{f}^2, \ldots, \tilde{f}^s$ are s models of the objective f built from sample points; and $w^1, w^2, \ldots, w^s$ are positive weights such that $\sum_{p=1}^{s} w^p = 1$. The main difficulty lies in the attribution of the weights that must reflect the quality of the models. To do so, weights can be attributed so that the error of the aggregate model will be minimal, thus introducing an optimization subproblem [3, 87]. Another approach is to compute an error metric $\mathcal{E}^p$ for each model $\tilde{f}^p$ and then attribute a weight w^p that is a function of $\mathcal{E}^p$ [16, 37, 63]. In this type of strategy, an error metric must be chosen first. Common metrics for this purpose are statistical measures with cross-validation like root mean square error (RSME) and predicted residual sum of squares (PRESS) that take into account the gaps between the values of the models and those of the true objective. In [16], the authors propose an error metric that is not statistical but is rather a measure of a model's capacity to rank points in the same order as the objective would do. The rationale for this latter metric is that in a BBO context a good model does not necessarily approximate well the values of

the true function but is rather capable of discriminating candidate points and telling apart promising ones.

Once an error metric is chosen, weights must be attributed accordingly. For instance, in [37], the authors propose the three following options:

$$\begin{aligned} w^p &\propto \mathcal{E}^{\text{tot}} - \mathcal{E}^p \\ \text{or } w^p &\propto \mathbb{1}_{\mathcal{E}^p = \mathcal{E}^{\min}} \\ \text{or } w^p &\propto (\mathcal{E}^p + \alpha \mathcal{E}^{\text{av}})^\beta \end{aligned}$$

where $\mathcal{E}^{\text{tot}}$ is the total error of all models, $\mathcal{E}^{\min}$ is the minimal error, $\mathcal{E}^{\text{av}}$ is the average error, and $\alpha < 1$ and $\beta < 0$ are adjustable parameters.

Bayesian optimization

In general, models may be trusted in areas where the true functions have been sufficiently sampled. But the further away from these areas, the less accurate are the models. In Bayesian optimization, the objective function is interpreted as a stochastic process: an *a priori* distribution $P[f]$ is assumed, then with the set of sample points $\mathbb{X}$ and a likelihood model $P[\mathbb{X} \mid f]$, an *a posteriori* distribution $P[f \mid \mathbb{X}]$ is built thanks to Bayes' rule. The latter distribution completely characterizes the stochastic process, including its mean and variance. Consequently, for any point x of the search space, a stochastic model not only produces a prediction $\mu(x)$ but also a measure of uncertainty on that prediction $\sigma(x)$. If properly exploited, this uncertainty enables to explore areas in which the model confesses to be unreliable, instead of spending the entire budget on restricted areas. This is called the compromise between exploration and exploitation. Commonly used stochastic models are generalized linear models [65], Gaussian processes [67], and dynamic trees [77].

The compromise between exploration and exploitation is then realized with an acquisition function. A simple example is the upper confidence bound (UCB) [76] that takes into account the most optimistic value, i.e., minimizes $\mu(x) - \kappa\sigma(x)$. A more sophisticated instance is the probability of improvement (PI) [44] which is the probability that the objective decreases from the best known value at a given point. Finally, a very popular example is the expected improvement (EI) [58] that not only takes into account the probability of decrease but also the expected amplitude thereof.

Figure 4.1 shows an example of Gaussian process regression - also know as *kriging* - as well as the resulting expected improvement on a one-dimensional objective function. The dashed curve represents the objective $f : x \mapsto x \sin x$; the five dots are the sample points; the curve

interpolating the dots is the prediction $\mu : x \mapsto \mu(x)$; an the filled area represents the 95% confidence interval given at any point x by $[\mu(x) + 1.96\sigma(x),\ \mu(x) - 1.96\sigma(x)]$. The curve at the bottom represents - in a different scale - the expected improvement EI. The resulting candidate point maximizes EI and is indicated by the vertical dashed line. Maximizing EI is

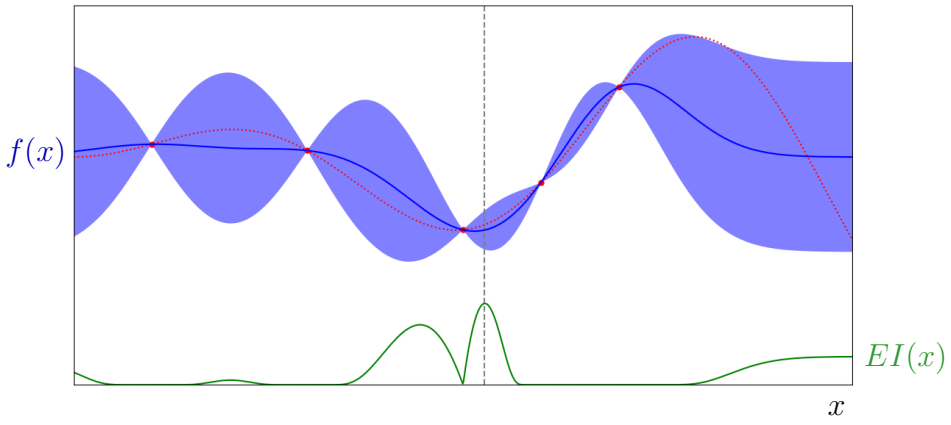


Figure 4.1 Kriging and expected improvement (EI) on $f : x \mapsto x \sin x$.

a method introduced by Jones et al. [45] that is efficient and easy to grasp. More elaborated techniques have since been developed. Talgorn et al. [79] propose various formulations for the surrogate subproblem that use multiple acquisition functions at once.

For an extended literature review on Bayesian optimization, see [40, 75].

4.2.2 The MADS algorithm

MADS [10] is a direct-search algorithmic framework that follows a search-poll paradigm in which the mandatory poll step guaranties the convergence and the optional search step gives room for flexible exploration techniques. In order to ensure convergence, every candidate point must lie on a *mesh* defined at Iteration k by

$$M^k = \left\{ x + \delta^k D y \ :\ x \in V^k,\ y \in \mathbb{N}^{n_D} \right\} \subset \mathbb{R}^n$$

where $V^k \subset \mathbb{R}^n$ is the *cache*, i.e., the set of all evaluated points up to Iteration k; $\delta^k > 0$ is the *mesh size parameter*; and D is a fixed matrix of $\mathbb{R}^{n \times n_D}$, the columns of which represent n_D directions of $\mathbb{R}^n$. Before the algorithm starts, V^0 is the set of one or more initial points provided by the user.

The search step enables to use various strategies to explore the space of variables. When the

search is unsuccessful, i.e., when no better solution is found, a poll step is launched. Every candidate point generated during the poll step must lie within a frame centred around the incumbent solution x^k and which size is parameterized by the *poll size parameter* $\Delta^k \geq \delta^k$.

At the end of an iteration, the mesh and poll size parameters are updated depending on the outcome. If the iteration is unsuccessful both are increased, and conversely, if the iteration is successful both are decreased in such a way that the set of possible directions during the poll gets richer. In this work, OrthoMADS [2] is used to deterministically generate $2n$ orthogonal directions at the poll step. A high-level description of MADS is given Algorithm 6.

Algorithme 6 : The Mesh Adaptive Direct-Search algorithm (MADS).

0. Initialization
$V^0 \subset \mathbb{R}^n$: set of starting points
$\Delta^0 \geq \delta^0 > 0$: initial mesh and poll size parameters
$k \leftarrow 0$: iteration counter

1. Search (optional)
Evaluate a finite set of points included in the mesh M^k.
If the search if successful, go to 3, otherwise go to 2.

2. Poll
Evaluate a finite set of points included in the poll frame.

3. Update parameters
Update the cache V^{k+1} with the newly sampled points.
Update the mesh and poll size parameters δ^{k+1} and Δ^{k+1}.
Increase the iteration counter $k \leftarrow k + 1$ and go to 1.

The algorithm stops either when the poll size parameter falls under a given threshold or when the prescribed budget of function evaluations is spent. Using the Clarke nonsmooth calculus [25], one can prove that under some mild assumptions on the smoothness of the problem, the MADS algorithm globally converges to a solution satisfying local optimality conditions provided that all candidate points lie on the mesh M^k. The interested reader may refer to [10] and [14, Chapter 8].

In the MADS context, surrogates can be used to find new candidate points during the search step. For instance, the few best solutions of a subproblem that uses only surrogates might be some promising candidate points. Several works have tackled the incorporation of surrogates in MADS like quadratic models [26], treed Gaussian processes [38], LOWESS models [78], hybrid models between static surrogates and dynamic models [9], or ensembles of model [16].

4.3 Quantifying uncertainty with ensembles of models

Ensembles of models enable to combine several models in the hope of taking advantage of each of them. However, this technique creates an aggregate model that can produce a prediction at any point but not an uncertainty on that prediction, thus prohibiting any Bayesian-like approach. Yet, because several models are useful to describe a single function, it means that their predictions are not identical. Consequently, there should be areas in the search space where the predictions show discrepancies, thus resulting in some form of uncertainty that is not apparent in the aggregate prediction. The proposed approach is precisely to catch the discrepancies between the models in order to produce a measure of uncertainty.

The idea that the disparity of the models' predictions can be used is tackled in [37] where the uncertainty at a given point x is produced with the standard deviation between the predictions defined by

$$\sigma(x) = \left(\frac{\sum_{p=1}^{s} \left(\tilde{f}^p(x) - \bar{f}(x) \right)^2}{s-1} \right)^{\frac{1}{2}}$$

where $\bar{f}(x) = \sum_{p=1}^{s} \tilde{f}^p(x)/s$. This metric quantifies the gaps between the values of the models, however, as it was said earlier, in a BBO context the actual values matter less than the variations of the models. For instance, the two following models possess significantly different values: $\tilde{f}^1$ and $\tilde{f}^2 = \tilde{f}^1/10 + 20$. Yet, their variations are the same, i.e., when $\tilde{f}^1$ increases, $\tilde{f}^2$ increases too and reciprocally, and therefore they have the same optima. In this case the uncertainty shall be minimum since using either model will yield the same candidate points. Now the two following models differ: $\tilde{f}^1$ and $\tilde{f}^3 = -\tilde{f}^1$, but in addition their variations will be opposite so that their optima will certainly be different. In this case, the uncertainty shall be maximum even though the actual standard deviation between $\tilde{f}^1$ and $\tilde{f}^3$ might be less than between $\tilde{f}^1$ and $\tilde{f}^2$. In light of this, a measure of uncertainty suited to BBO should rather take into account the variations of the models in the form of some local correlation. In [60] and [81], a correlation coefficient is built between a high-fidelity surrogate $\tilde{f}^{\text{high}}$ and a low-fidelity model $\tilde{f}^{\text{low}}$

$$r = \frac{\sum_{j=1}^{M} \left(\tilde{f}^{\text{high}}\left(x^{(j)}\right) - \bar{f}^{\text{high}} \right) \left(\tilde{f}^{\text{low}}\left(x^{(j)}\right) - \bar{f}^{\text{low}} \right)}{\sqrt{\sum_{j=1}^{M} \left(\tilde{f}^{\text{high}}\left(x^{(j)}\right) - \bar{f}^{\text{high}} \right)^2} \sqrt{\sum_{j=1}^{M} \left(\tilde{f}^{\text{low}}\left(x^{(j)}\right) - \bar{f}^{\text{low}} \right)^2}}$$

where $\{x^{(j)}\}_{j \in \{1,2,\ldots,M\}}$ is a set of $M \geq n$ points sampled locally around an area of interest; and $\bar{f}^{\text{high}}$ and $\bar{f}^{\text{low}}$ are the average values of $\tilde{f}^{\text{high}}$ and $\tilde{f}^{\text{low}}$ on this set of points, respectively.

For the reasons aforementioned, this quantity is more relevant than the standard deviation in BBO.

4.3.1 A new expression for the uncertainty

In the proposed approach, this idea of correlation between models is exploited to produce an expression of the uncertainty at a given point x. Two alternatives are built: a smooth and a nonsmooth uncertainties. In addition each alternative is declined into two versions: an uncertainty dedicated to the objective and another one dedicated to the constraints.

Smooth uncertainty for the objective

The simplex gradient [46] of a function f at point x, denoted by $\nabla_S f(x)$, is the gradient of the linear model of f at x. Computing the simplex gradient around a given point x requires the evaluation of f on a simplex, i.e., a set of $n+1$ affinely independent points, around x. The correlation between two models can be reinterpreted geometrically. For two models $\tilde{f}^p$ and $\tilde{f}^q$, the cosine between their simplex gradients at a given point x is defined by

$$\cos\left\langle \nabla_S \tilde{f}^p(x),\ \nabla_S \tilde{f}^q(x) \right\rangle = \frac{\nabla_S \tilde{f}^p(x)^\top \nabla_S \tilde{f}^q(x)}{||\nabla_S \tilde{f}^p(x)||_2 \times ||\nabla_S \tilde{f}^q(x)||_2} \ .$$

The larger the cosine, the more correlated the models around x. With this notion in mind, the uncertainty between two models $\hat{\sigma}_{p,q}$ can be produced as an inversely proportional function of the cosine

$$\hat{\sigma}_{p,q}(x) := \frac{1}{2}\left(1 - \cos\left\langle \nabla_S \tilde{f}^p(x),\ \nabla_S \tilde{f}^q(x) \right\rangle\right) \ . \tag{4.2}$$

When the models are highly correlated, the cosine is close to 1 so that the uncertainty is close to its minimum 0. When the models are poorly correlated, the cosine is closer to 0 and the uncertainty increases to 0.5. And when the models are anti-correlated, the cosine is close -1 so that the uncertainty reaches its maximum 1. The choice of a simplex is left at the discretion of the user. A small simplex around x will yield a simplex gradient that is a good approximation of the true gradient for smooth functions, but a wider simplex will have a smoothing effect that can be appreciable with nonsmooth or noisy functions. In the current context, the simplex gradient acts as a simple surrogate for the true gradients of the models $\tilde{f}^p$, $p \in \{1, 2, \ldots, s\}$ that are not always easy to obtain. However, using the true gradients if available might be equally efficient. See Appendix A for the practical construction of the simplex used in this work.

The generalization of this expression to more than two models will be described after the

other versions of uncertainties are introduced.

Nonsmooth uncertainty for the objective

An alternative for the uncertainty that does not require the computation of simplex gradients is proposed. It requires a positive spanning set of directions $\mathcal{D}$, i.e., a set of at least $n+1$ vectors of $\mathbb{R}^n$ such that any point of $\mathbb{R}^n$ can be written as a positive linear combination thereof [30]. The nonsmooth alternative is defined by

$$\hat{\sigma}_{p,q}(x) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \mathsf{xor}\Big(\tilde{f}^p(x+d) < \tilde{f}^p(x) \ , \ \tilde{f}^q(x+d) < \tilde{f}^q(x) \Big) \tag{4.3}$$

where $\mathsf{xor}(\cdot\ ,\cdot)$ is the exclusive or logical operator. For each direction $d \in \mathcal{D}$, the uncertainty increases if the models predict contradictory trends, i.e., if model $\tilde{f}^p$ increases from x to $x+d$ while $\tilde{f}^q$ decreases, or conversely. The term $1/|\mathcal{D}|$ scales the sum between 0 and 1 so that the uncertainty will not be influenced by the number of directions in $\mathcal{D}$. Here again, the choice of a positive spanning set is left at the discretion of the user. Not only can the size of the directions vary as with the simplex, but also the number of directions can increase in order to explore the surroundings of x better. See Appendix A for the practical construction of the positive spanning set used in this work.

Smooth uncertainty for the constraints

The expressions for the uncertainty proposed in Equations (4.2) and (4.3) are suited for the objective f since they take into account variations of the models. However, when handling constraints, the key information is the sign of the function rather than whether it increases or not. If two models $\tilde{c}_j^p$ and $\tilde{c}_j^q$ of the same constraint c_j are available, the uncertainty shall increase when one model or the other tends towards 0, and increase even more when their signs are opposite, meaning that their predictions on the feasibility are contradictory. Hence the following expression

$$\hat{\sigma}_{p,q}(x) = \mathrm{sigm}\left(-\tilde{c}_j^p(x) \times \tilde{c}_j^q(x) \right) \tag{4.4}$$

where $\mathrm{sigm}(\cdot)$ is the sigmoid function. It acts as an activation function that increases as the product of $\tilde{c}_j^p$ and $\tilde{c}_j^q$ decreases. This uncertainty also ranges from 0 to 1.

Nonsmooth uncertainty for the constraints

Here again, a nonsmooth alternative is proposed to the smooth uncertainty for the constraints. It uses the logical operator xor to indicate whether the two models $\tilde{c}_j^p$ and $\tilde{c}_j^q$ predict

the same feasibility result at a given point x or not

$$\hat{\sigma}_{p,q}(x) = \mathsf{xor}\left(\tilde{c}_j^p(x) \leq 0 \ , \ \tilde{c}_j^q(x) \leq 0\right) \tag{4.5}$$

Generalization to an arbitrary number of models

Expressions (4.2), (4.3), (4.4) and (4.5) consider two models of the objective or a constraint. Ensembles of models usually comprise more than two models though, hence the need for a general expression that can consider an arbitrary number of models. In addition, this general expression must take into account the weights w^p, $p \in \{1, 2, \ldots, s\}$, which reflect the quality of the models. Just as in the prediction defined in Equation (4.1), the good models should have a strong influence in the determination of the uncertainty whereas the poor models should not. The following quantity meets those requirements

$$\left(\sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q \times \hat{\sigma}_{p,q}(x)\right) \Big/ \sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q \tag{4.6}$$

The ratio in (4.6) considers all the possible pairs of models once. For each pair $(p, q) \in \{1, 2, \ldots, s\}^2$ such that $p \neq q$, the uncertainty $\hat{\sigma}_{p,q}(x)$ stemming from the models $\tilde{f}^p$ and $\tilde{f}^q$ at point x is weighted by the product of the corresponding weights w^p and w^q. Consequently, the better the models, the more $\hat{\sigma}_{p,q}(x)$ will weight up in the total uncertainty at point x. Then the sum on all pairs of models is normalized by $1/\sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q$ so that the result does not depend on the number of models s. Since the four versions of $\hat{\sigma}_{p,q}$ range from 0 to 1, this ratio applies to any case: objective and constraint versions, smooth and nonsmooth alternatives.

At this point the uncertainty takes into account an arbitrary number of models and also the weights as required. But the ratio in (4.6) is between 0 and 1 by construction, and therefore it is most likely not at the right scale for the problem at hand. A final step is to multiply by a factor $\alpha > 0$ that scales the ratio in a relevant way. For this purpose, $\alpha = 10 \times \mathrm{Var}(g(V))$ was chosen, where g is either the objective or a constraint; $g(V) = \left\{g\left(x^{(1)}\right), g\left(x^{(2)}\right), \ldots, g\left(x^{(N_s)}\right)\right\}$ is the set of already sampled values of the function g; and 10 is a factor that empirically gave better results. This choice is motivated by the fact that the problem's scale can only be known through the true function's values. The final expression of the uncertainty is

$$\hat{\sigma}(x) = \alpha \frac{w^\top \Sigma(x) w}{w^\top T w} \tag{4.7}$$

where the ratio (4.6) has been rewritten in a more compact form with $\Sigma(x) \in \mathbb{R}^{s \times s}$ being

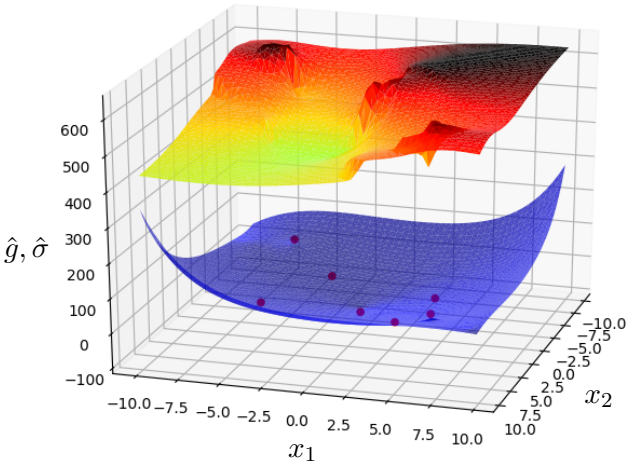
the upper triangular matrix such that $[\Sigma(x)]_{p,q} = \hat{\sigma}_{p,q}(x)$ if $p < q$ and 0 otherwise; $T \in \mathbb{R}^{s \times s}$ being the upper triangular matrix such that $[T]_{p,q} = 1$ if $p < q$ and 0 otherwise; and w being the vector of weights $[w^1, w^2, \ldots, w^s]^\top$.

The different uncertainties are illustrated in Figure 4.2. Seven points have been sampled in $[-10, 10]^2$ from an unknown function g that takes two input variables x_1 and x_2. On each subfigure, the bottom surface is the aggregate prediction $\hat{g}$ resulting from eleven different polynomial and RBF models, and the top surface, shifted up for readability, is the uncertainty on that prediction which becomes darker as it increases. The weights of the models w^p, $p \in \{1, 2, \ldots, s\}$, are attributed as described in Section 4.3.2. In Figures 4.2a and 4.2b the function is interpreted as the objective whereas in Figures 4.2c and 4.2d the same function is interpreted as a constraint, resulting in significantly different uncertainties. In addition, in Figures 4.2c and 4.2d, the points close to the assumed border of the constraint, i.e., where the prediction is close to zero, were darkened in order to better understand the uncertainty.

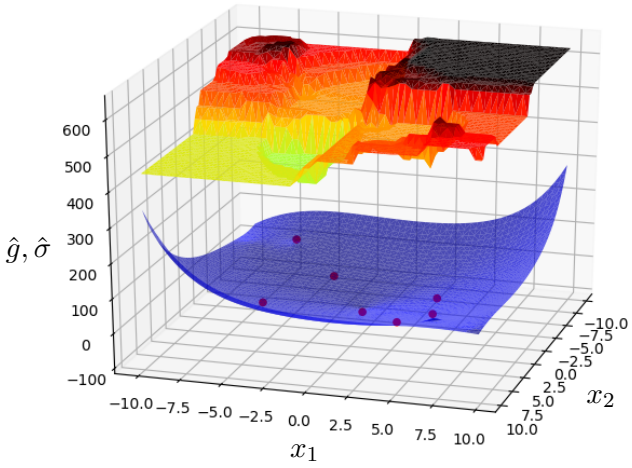
Comparatively, Figure 4.3 shows a GP model's prediction and uncertainty fit on the same sample points. It can be noticed that the uncertainty is lower close to the sample points and increases with the distance to them, which is expected with a GP model. The same observation cannot be made in Figure 4.2, especially in Figures 4.2c and 4.2d where the uncertainty is higher close to the border of the constraint.

4.3.2 Error metric and weight attribution

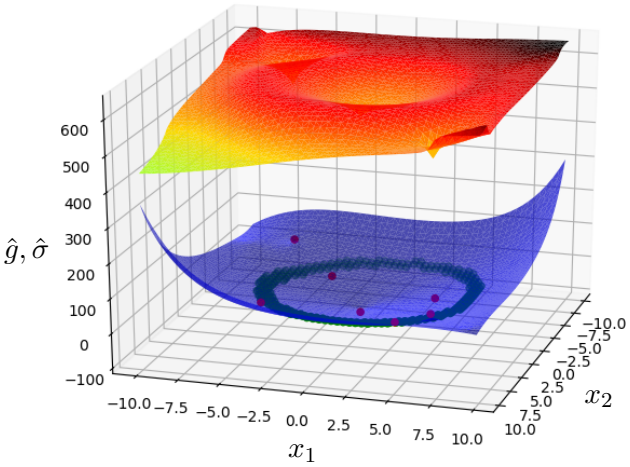
In this work, the strategy to assign weights is to compute an error metric $\mathcal{E}^p$ for each model $\tilde{f}^p$ and then attribute a weight w^p that is a function of $\mathcal{E}^p$. The error metric chosen is the order error cross-validation metric (OECV) [16] mentioned earlier and denoted by $\mathcal{E}_{OECV}$. Broadly speaking, it measures a model capacity to rank points in the same order as the actual objective would do, or, for that matter, its capacity to predict the same feasibility result as an actual constraint would do. When the metric $\mathcal{E}^p_{OECV}$ is computed for every model $\tilde{f}^p$, $p \in \{1, 2, \ldots, s\}$, the weights can be attributed. Assigning a weight of 1 to the best model and 0 to the others is the choice made in [16]. However, in the present work there must be at least two strictly positive weights otherwise the ratio (4.6) is a division by zero and has no meaning. The approach chosen instead is to select the N_{best} models that have the smallest error metrics and to assign to them a weight proportional to the metric. Formally, if $I \subset \{1, 2, \ldots, s\}$ is the subset of the selected models indices, then $w^p \propto \mathcal{E}^{\text{tot}}_I - \mathcal{E}^p$ if $p \in I$ and $w^p = 0$ otherwise, where $\mathcal{E}^{\text{tot}}_I$ is the total error of the selected models. The weights are then normalized so that $\sum_{p \in I} w^p = 1$. If more than N_{best} models have an error metric that is equal to the best metric, all of them will be selected and be assigned an equal



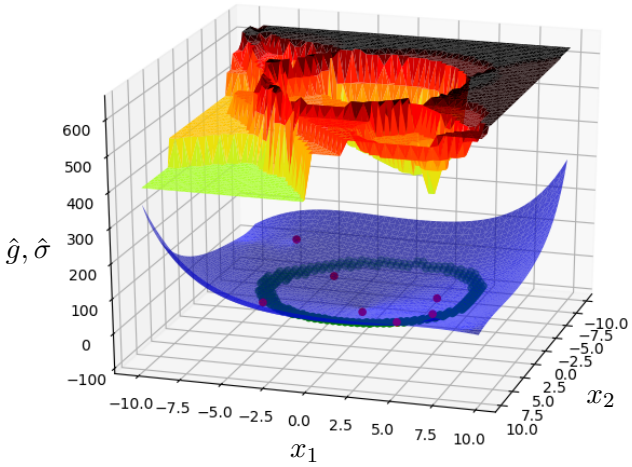
(a) Objective version - smooth alternative.



(b) Objective version - nonsmooth alternative.



(c) Constraint version - smooth alternative.



(d) Constraint version - nonsmooth alternative.

Figure 4.2 The four uncertainties on the same sample set. Figures 4.2a and 4.2b correspond to the the smooth and nonsmooth alternatives of the objective version, respectively (Equations (4.2) and (4.3)). Figures 4.2c and 4.2d correspond to the the smooth and nonsmooth alternatives of the constraint version, respectively (Equations (4.4) and (4.5)).

weight. Preliminary tests showed that $N_{\text{best}} = 3$ and $N_{\text{best}} = 4$ were appropriate values for the smooth and nonsmooth alternatives, respectively.

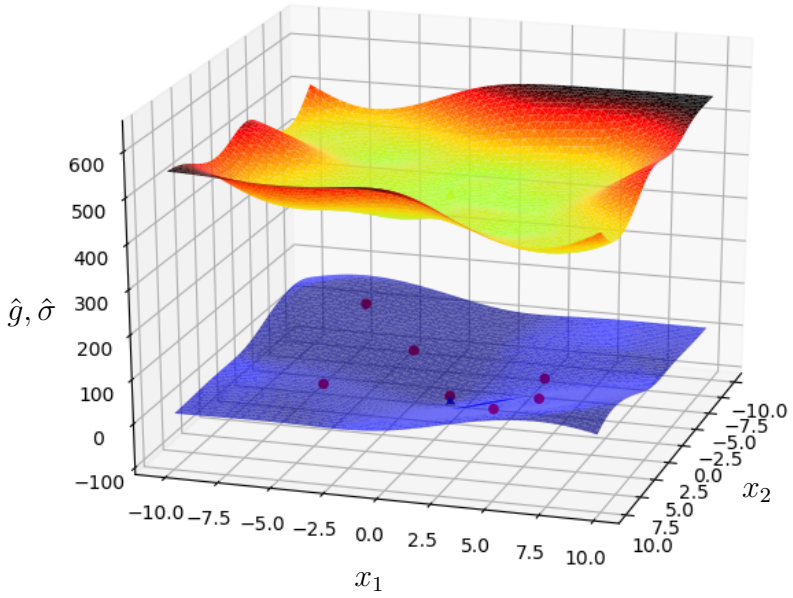


Figure 4.3 Prediction and uncertainty of Gaussian processes.

4.3.3 Incorporation into the MADS algorithm

The MADS algorithm offers an important flexibility through its search step. Many works have already included model-based subproblems (SP) into the search step as described earlier. The proposed implementation falls within this category. At each iteration, a surrogate problem is solved during the search step and the best solution is used as the next candidate point for the true problem. Having said that, there are many ways to design a surrogate SP.

In [79], eight different formulations for SP are proposed, denoted by SP1 through SP8. They are specifically designed to take advantage of the double information that stochastic models provide, i.e., the prediction and the uncertainty. These formulations tackle general constraints and involve the following statistical measures:

$$\begin{cases} \text{Expected improvement [45]:} & \text{EI}(x) & = & \mathbb{E}[\max(f_{min} - f(x), 0)] \\ \text{Probability of feasibility [79]:} & \text{P}(x) & = & \mathbb{P}[c_j(x) \leq 0, j = 1, 2, \ldots, m] \\ \text{Probability of improvement [44]:} & \text{PI}(x) & = & \mathbb{P}[f_{min} > f(x)] \end{cases}$$

Then some other measures are derived from these quantities: the *expected feasible improvement* $\text{EFI}(x) = \text{EI}(x)\text{P}(x)$, the *probability of feasible improvement* $\text{PFI}(x) = \text{PI}(x)\text{P}(x)$ and the *uncertainty on the feasibility* $\mu(x) = 4\text{P}(x)(1 - \text{P}(x))$. In the formulations, all these

measures are arranged in different ways in order to highlight various properties of the sample points. The eight formulations of SP are given in Appendix B.

With stochastic models, the probability distribution at any point x is known and used to practically compute EI(x), P(x) and PI(x). However, an aggregate model extended with the proposed uncertainty, although inspired by the stochastic modelling philosophy, has no probabilistic foundation. More specifically, there is no cumulative distribution function available at a given point $\mathbb{P}[g(x) < g_0]$, for all $g_0 \in \mathbb{R}$. Consequently, the statistical quantities defined above are not applicable as such. To address this issue, P, PI and EI are replaced by substitutes $\widetilde{\mathrm{P}}$, $\widetilde{\mathrm{PI}}$ and $\widetilde{\mathrm{EI}}$ that are inspired from the case when the stochastic model yields at any point x a value that follows a normal distribution $\mathcal{N}\Big(\hat{y}(x), \hat{\sigma}^2(x)\Big)$, which is the case of GPs.

When $\hat{c}_j(x) \sim \mathcal{N}\Big(\hat{y}_j, \hat{\sigma}_j^2\Big)$, the expression of P becomes

$$\mathrm{P}(x) = \prod_{j=1}^{m} \Phi\left(-\frac{\hat{y}_j}{\hat{\sigma}_j}\right)$$

where Φ is the standard normal cumulative distribution function. Here the product implies that the constraints are assumed to be independent from each other, which might be incorrect but it is the best available approximation in a BBO context. The proposed adaptation is

$$\widetilde{\mathrm{P}}(x) = \prod_{j=1}^{m} \mathrm{sigm}_\lambda \quad -\frac{\hat{y}_j}{\hat{\sigma}_j}\Bigg)$$

where sigm_λ is the sigmoid function of parameter λ, i.e., $\mathrm{sigm}_\lambda(x) = \mathrm{sigm}(\lambda x)$. Both Φ and sigm_λ tend to 1 when the ratio $-\hat{y}_j/\hat{\sigma}_j$ tends to $+\infty$, i.e., either when $\hat{y}_j$ is highly negative or when the uncertainty is low for whatever negative value of $\hat{y}_j$, which in both cases means that the constraint c_j is most likely satisfied. They also tend to 0 when the ratio $-\hat{y}_j/\hat{\sigma}_j$ tends to $-\infty$, i.e., either when $\hat{y}_j$ takes high values or when the uncertainty is low for whatever positive value of $\hat{y}_j$, which in both cases means that the constraint c_j is most likely not satisfied. $\lambda = 2$ produces the closest approximation of Φ but choosing other values enable to control the shape of the function. Preliminary tests showed that $\lambda = 3$ and $\lambda = 1$ were interesting values for the smooth and nonsmooth alternatives, respectively.

As for PI, when $\hat{f}(x) \sim \mathcal{N}\Big(\hat{y}, \hat{\sigma}^2\Big)$, the expression becomes

$$\mathrm{PI}(x) = \Phi\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where f_{min} is the best know value of the objective. The proposed alternative is

$$\widetilde{\mathrm{PI}}(x) = \mathrm{sigm}_\lambda\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

Here again, Φ and sigm_λ have the same behaviour but the parameter λ enables to control the shape of PI. The values chosen for the smooth and nonsmooth alternatives are $\lambda = 0.1$ and $\lambda = 0.5$, respectively.

Finally, when $\hat{f}(x) \sim \mathcal{N}\left(\hat{y}, \hat{\sigma}^2\right)$, the expression of EI becomes

$$\mathrm{EI}(x) = (f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right) + \hat{\sigma}\phi\quad\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where ϕ is the standard normal density function. This expression is intimately related to the Gaussian aspect of the model an therefore is not *a priori* suited for non-Gaussian models, let alone models that are not truly stochastic. However, it possesses interesting properties that are independent from the Gaussian nature of the model and that can be seen as essential to the very notion of expected improvement. Firstly, for a fixed $\hat{\sigma}$, EI increases when $\hat{y}$ decreases, which is judicious in a minimization context since an improvement means a lower value of the objective. Subsequently, EI is almost positively proportional to $\hat{y}$ when $\hat{y}$ tends to $-\infty$, which is appropriate for the same reason. Moreover, EI tends to 0 when $\hat{y}$ tends to $+\infty$, which is also relevant because when the prediction becomes extremely high, no improvement can be expected. In addition, when $\hat{y}$ gets closer to f_{min}, EI becomes almost proportional to $\hat{\sigma}$, meaning that when the prediction does not improve the objective (i.e., $\hat{y} \simeq f_{min}$) the expected improvement mostly relies on the uncertainty. Then, for a fixed $\hat{f}$, EI increases in $\hat{\sigma}$ and is almost proportional to $\hat{\sigma}$ when $\hat{\sigma}$ tends to $+\infty$, which is sensible since for a given prediction the higher the uncertainty, the larger the potential improvement. Finally, when $\hat{\sigma}$ tends to 0, the behavior of EI depends on the values of f_{min} and $\hat{y}$: if $f_{min} \geq \hat{y}$, then EI tends to $f_{min} - \hat{y}$, and if $f_{min} < \hat{y}$, then EI tends to 0, meaning that when the uncertainty is low, EI mostly relies on the comparison between f_{min} and the prediction $\hat{y}$. Taking into account these considerations, the proposed adaptation for EI is very close to the actual EI and is defined by

$$\widetilde{\mathrm{EI}}(x) = (f_{min} - \hat{y})\ \mathrm{sigm}_\lambda\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right) + \hat{\sigma}\gamma\quad\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where $\gamma(t) = e^{-t^2/2}$. Here $\lambda = 1$ was chosen. The functions ϕ and γ only differ by a factor $1/\sqrt{2\pi}$ and the reason for choosing γ instead of ϕ is that this factor is no more

justified without an actual stochastic model that produces normal distributions. Moreover, preliminary tests showed that the proposed uncertainty seemed on average lower than the uncertainty provided by a kriging model. The terms P, PI and EI were then replaced by $\widetilde{\mathrm{P}}$, $\widetilde{\mathrm{PI}}$ and $\widetilde{\mathrm{EI}}$ in the formulations of SP.

Algorithme 7 : The MADS algorithm with aggregate models.

0. Initialization

$\mathrm{SP} \in \{\mathrm{SP}_1, \mathrm{SP}_2, \ldots, \mathrm{SP}_8\}$	: surrogate subproblem formulation
$\tilde{g}^1, \tilde{g}^2, \ldots, \tilde{g}^s$	: choice of models for the objective and the constraints
$V^0 \subset \mathbb{R}^n$	: set of starting points
$\Delta^0 \geq \delta^0 > 0$	: initial mesh and poll size parameters
$k \leftarrow 0$	: iteration counter

1. Models and weights update

Build or update $\tilde{f}^1, \tilde{f}^2, \ldots, \tilde{f}^s$ using the values of f in V^k
Update $w^1, w^2, \ldots, w^s$ using the OECV metric for the objective
Build or update $\tilde{c}_j^1, \tilde{c}_j^2, \ldots, \tilde{c}_j^s$ using the values of c_j in V^k, for $j \in \{1, 2, \ldots, m\}$
Update $w_j^1, w_j^2, \ldots, w_j^s$ using the OECV metric for the constraints, for $j \in \{1, 2, \ldots, m\}$

2. Search

Solve SP to find the best solution x_{SP}^k
Project x_{SP}^k onto the mesh M^k
Evaluate the resulting point with the true problem

3. Standard poll

4. Standard parameters update

Algorithm 7 summarizes the incorporation of extended aggregate models in MADS. First, one formulation must be chosen among {SP1, SP2, ..., SP8}. At iteration k, the best solution found for SP, denoted by x_{SP}^k, is projected onto the mesh M^k and is used as the candidate point of the search step. The freshly evaluated points are then added to the cache V^k so that the models and the weights will be adjusted accordingly before iteration $k+1$ begins. The resulting algorithm benefits from the convergence results of MADS since all the candidate points lie on the mesh M^k.

4.4 Computational results

The proposed approach has been tested on seven analytical problems; two multi-disciplinary optimization (MDO) applications: the aircraft range problem and the simplified wing problem; and two simulation problems: solar1 and styrene. Version 4 of the NOMAD software [1,17] was used on a PC Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz on Linux. The aggregate models used the default selection of eighteen models comprised of polynomial response surfaces

of various degrees, kernel smoothing, modified radial basis functions as in [16], and closest neighbours. The competing quadratic and kriging models were also readily available in NOMAD. Due to the long running times required by the kriging models, the latter were only tested on the analytical problems and the aircraft range problem. Every instance of MADS in this work uses the last direction of success at the poll step [10]. Two other BBO solvers have been included in this study: DFN [32] and SHEBO [61]. The former innately handles general constraints whereas the latter is designed for problems with hidden constraints. Consequently, the problems were adapted in SHEBO so that a violated general constraint will be interpreted as a hidden constraint.

The interpretation of the results mostly relies on data profiles [59] which enable to compare multiple solvers on a given set of problems. Broadly speaking, the data profile of a solver indicates the proportion of problems solved to a given tolerance within a prescribed number of evaluations. Since SHEBO does not take a single starting point as an input, it is not fit for comparison to the other algorithms through data profiles. Section 4.4.6 provides tabular comparisons with SHEBO.

Unless otherwise specified, due to the randomness contained in MADS, every version thereof was run four times on each problem with a different seed for the random generator each time. Similarly, DFN enables to choose between the Halton and Sobol sequences so the two were tested on each problem and taken into account in the data profiles.

4.4.1 Analytical problems

The seven analytical problems are listed in Table 4.1 with the number of variables n and constraints m, whether the variables are bounded or not, and the number of starting points used. By taking into account the additional starting points for problems HS83, HS114 and MAD6, the total number of problems is fifteen. The evaluation budget is $1200(n+1)$.

Table 4.1 Description of the seven analytical problems.

#	Name	Source	n	m	Bounds	# starting points
1	G2	[12]	10	2	yes	1
2	HS19	[41]	2	2	yes	1
3	HS83	[41]	5	6	yes	4
4	HS114	[53]	9	4	yes	3
5	MAD6	[53]	5	7	no	4
6	PENTAGON	[53]	6	15	no	1
7	SNAKE	[11]	2	2	no	1

As in [79], the eight SP formulations were compared, and the following values were tested for the parameter λ when applicable: $\{0, 0.01, 0.1, 1\}$, thus resulting in twenty-three distinct formulations. When this parameter is involved in a formulation, it is denoted as a subscript, e.g., $\text{SP2}_{0.1}$. The purpose here is not to exhaustively compare the formulations with each other but rather to identify the bests formulations and compare their performances to the existing versions of NOMAD and to DFN.

The formulations with extended aggregate models were compared to NOMAD without any search step, referred to as "no search", NOMAD with a search step involving the minimization of quadratic models, referred to as "quad search", and DFN. It turns out that all formulations perform better than no search, confirming that the approach is valid and does not "waste" evaluations. However, quad search is most of the times as good as, and sometimes better than, the proposed approach, confirming the effectiveness of quadratic models on analytical functions. DFN presents heterogeneous performances. It found good solutions for three problems but performed poorly on the others, hence the low overall performance.

The best formulation found for the smooth alternative in terms of the proportion of problems solved within the budget is SP3_0 defined by

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & -\text{EI}(x) \\ \text{s.t.} \quad & \hat{c}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \end{aligned} \tag{SP3-EIσ}$$

As for the nonsmooth alternative, the best formulation is $\text{SP5}_{0.01}$ which consists in maximizing $\text{EFI}(x) + 0.01\hat{\sigma}_f(x)$. Figure 4.4 shows the data profiles of the five following algorithms: no search, quad search, SP3_0 with smooth uncertainty, $\text{SP5}_{0.01}$ with nonsmooth uncertainty, and DFN at variable tolerance: $\tau = 10^{-1}$, $\tau = 10^{-3}$, $\tau = 10^{-5}$ and $\tau = 10^{-7}$.

The profiles show that the performances of SP3_0 and $\text{SP5}_{0.01}$ are close to that of quad search for high tolerance. However, quad search becomes significantly better for low tolerance ($\tau = 10^{-7}$). These first results show that the extended aggregate models combined with the formulations manage to find good solutions as efficiently as quad search. However, quadratic models do so slightly faster in terms of the number of evaluations, and more importantly, they are especially accurate on analytical problems, thus resulting in superior performances at low tolerance.

Since extended aggregate models are meant to mimic and therefore supersede actual stochastic models, the comparison to the available kriging models in NOMAD is most appropriate. In [79], the authors recommend SP1 and SP2 with large values of λ when using stochastic models. For that reason, these two formulations have been tested with $\lambda = 0.1$ and $\lambda = 1$.

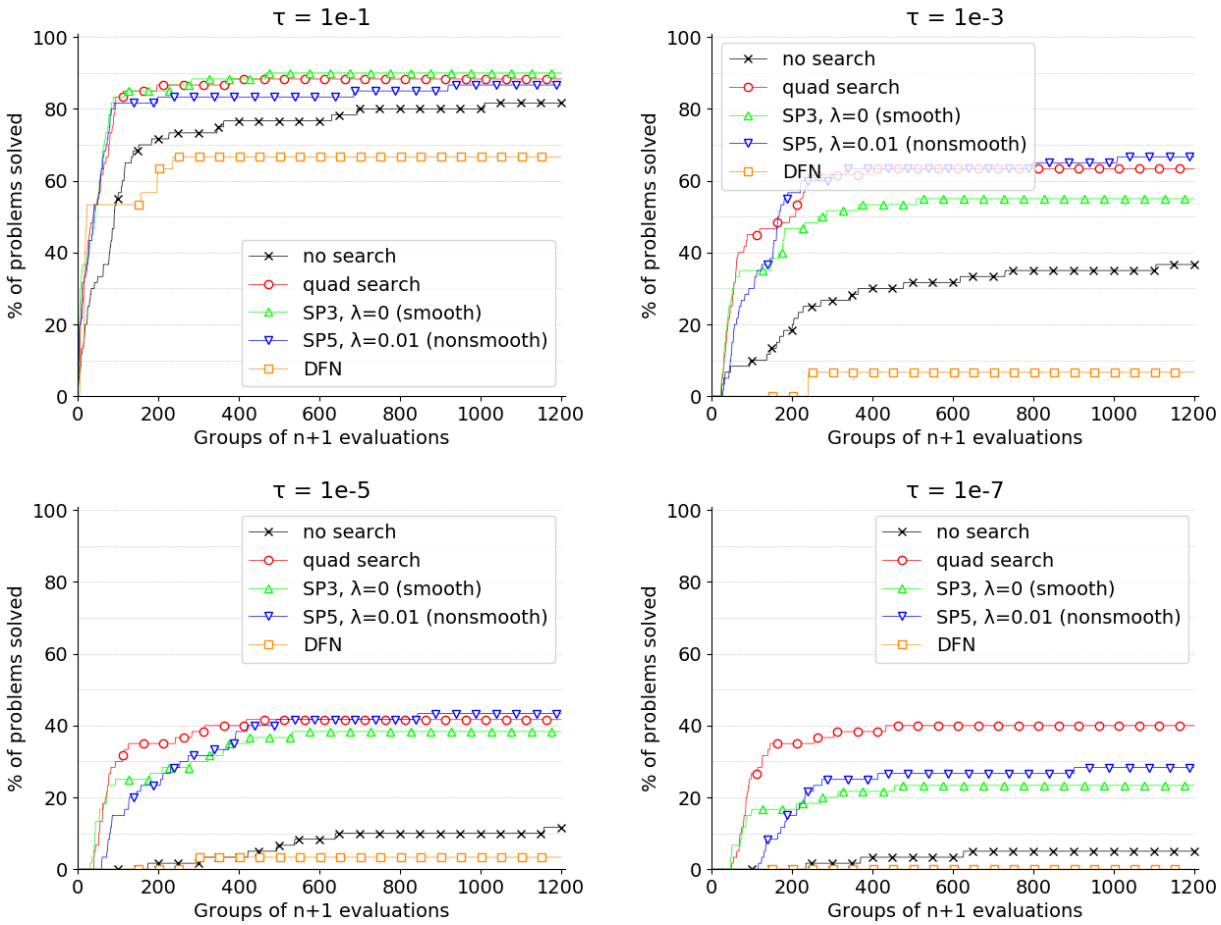


Figure 4.4 Data profiles. no search vs. quad search vs. $SP3_0$ with smooth uncertainty vs. $SP5_{0.01}$ with nonsmooth uncertainty vs. DFN on analytical problems.

On this set of problems, $SP2_{0.1}$ turns out to be the best formulation. The latter was therefore tested against $SP3_0$ with smooth uncertainty and $SP5_{0.01}$ with nonsmooth uncertainty, that is the best formulations seen above. The resulting data profiles in Figure 4.5 show that on the present set of problems extended aggregate models are as good as, or better than, the kriging alternative depending on the tolerance. In addition, due to the inversion of a covariance matrix that grows with the size of the sample set, the kriging models typically take minutes to tens of minutes to solve one problem, which is prohibitive when optimizing cheap functions with large budgets of evaluations. In comparison, the proposed approach and quad search typically take minutes and no search and DFN take seconds. As a result, replacing classic kriging models by extended aggregate models does not harm the performances in terms of the number of evaluations on the set of analytical problems, while improving the real optimization time. Overall, based on the present results, quad search must be favoured on cheap analytical problems.

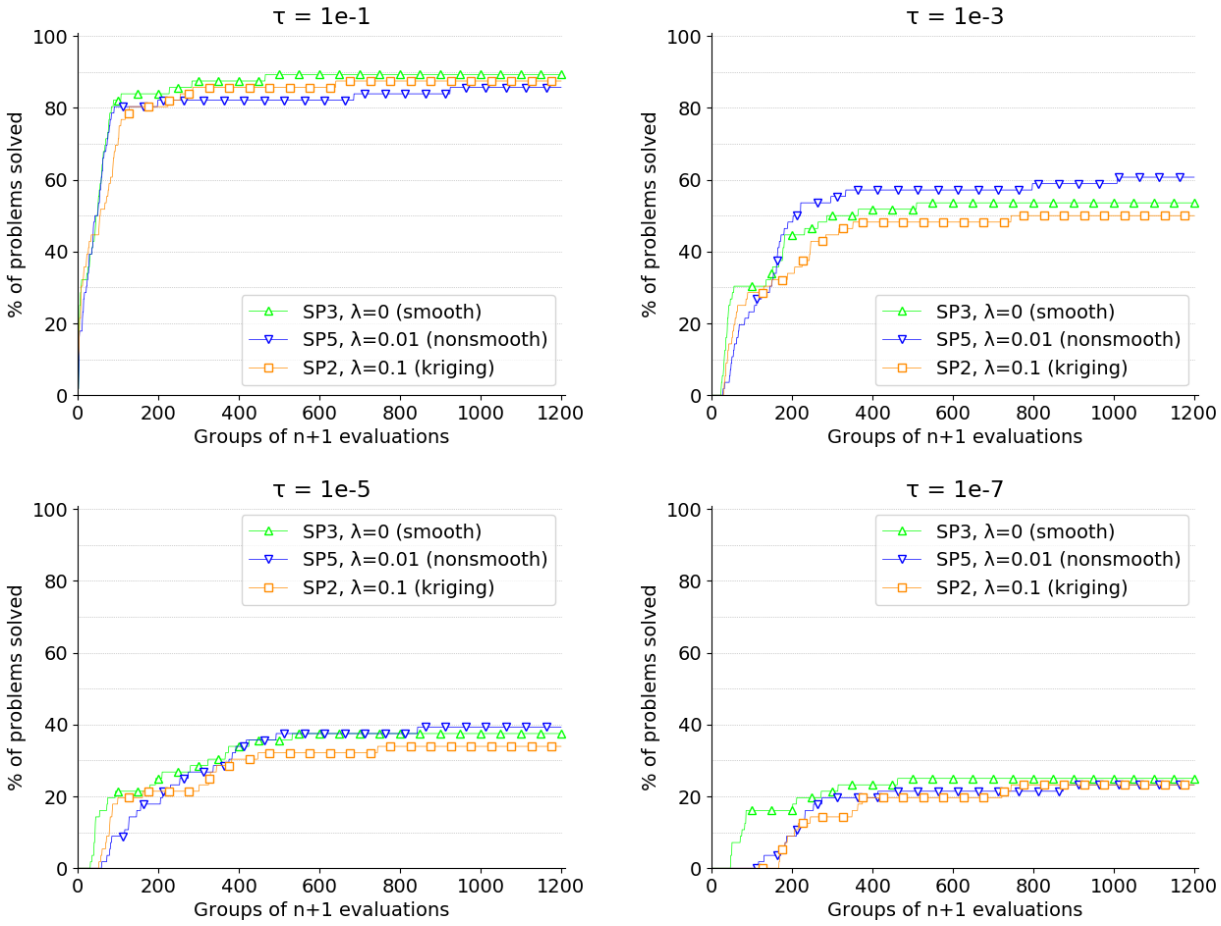


Figure 4.5 Data profiles. $SP3_0$ with smooth uncertainty vs. $SP5_{0.01}$ with nonsmooth uncertainty vs. $SP2_{0.1}$ with kriging models on analytical problems.

4.4.2 The aircraft range MDO problem

The aircraft range problem [49] is a multi-disciplinary optimization problem (MDO), meaning that it is combined of several interconnected disciplines so that the input of one discipline is the output of the others, and several cycles between them are necessary to stabilize the result. The aircraft range problem aims at maximizing the range of a supersonic business jet by considering aerodynamics, structure, and propulsion, under constraints of engine, performance and structure. The problem has $n = 10$ variables and $m = 10$ constraints. It is nonsmooth and has several local optima.

In order to test the algorithms truthfully, ten starting points were sampled with Latin hypercube sampling [55]. Here again, the 23 formulations were run with the two uncertainty alternatives in order to identify the best combinations and analyze their performances. The evaluation budget is $1000(n+1)$. The best formulation with the smooth alternative is $\text{SP3}_{0.1}$ defined by

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & -\text{EI}(x) - 0.1\hat{\sigma}_f(x) \\ \text{s.t.} \quad & \hat{c}_j(x) - 0.1\hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \end{aligned} \tag{SP3-EIσ}$$

and the best formulation identified with the nonsmooth alternative is SP8 which consists in maximizing $\text{PFI}(x)$.

This time, each of the aforementioned formulations is compared individually not only to quad search, but also to the same formulation with kriging models instead. This choice is motivated by the fact that the aircraft range problem is a real-world simulation-based problem that takes significantly more time to compute than the previous analytical problems, and therefore computing expensive kriging models might be worth the trade-off. In this section, no search is not shown in the interest of readability, and DFN neither because it performed poorly and its data profiles were flat. Figures 4.6a and 4.6b show the data profiles of $\text{SP3}_{0.1}$ with smooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively; and Figures 4.6c and 4.6d show the data profiles of SP8 with nonsmooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively. All the algorithms presented equivalent performances for high tolerances ($\tau = 10^{-1}$ and $\tau = 10^{-3}$) and consequently the data profiles did not reveal significant difference between the solvers with tolerances under $\tau = 10^{-5}$, meaning that all the formulations manage to reach a good solution.

For every above-mentioned formulation, the performance is relatively comparable to that of quad search for tolerance $\tau = 10^{-7}$. However, for $\tau = 10^{-9}$, the proposed extended aggregate models coupled with the right formulations turn out to be significantly better than both quad search and their kriging counterpart, i.e., the same formulations with kriging models

instead. It can be noticed that the smooth alternative eventually solves more problem, but the nonsmooth one solves problems faster. In Figure 4.6b, $\text{SP3}_{0.1}$ with smooth uncertainty solves 87.5% of the problems at tolerance $\tau = 10^{-9}$ with the allocated budget, while in Figure 4.6d SP8 with nonsmooth uncertainty only solves 77.5% of the problems at the same tolerance. However, the latter takes only $300(n+1)$ evaluations to do so, while the former has only solved 50% of the problems after the same number of evaluations. This trend has been observed on this problem with all the other formulations not presented in this paper. The higher achievements of the smooth uncertainty over the long run may be attributed to its intrinsically rich range of values, while the relative rapidity of the nonsmooth uncertainty may be the result of a more aggressive behaviour that helps find a good solution faster.

It could be argued that comparing kriging models within the best formulations found for extended aggregate models, i.e., $\text{SP3}_{0.1}$ and SP8, is not fair since the former might perform poorly on these formulations but yield better results on others. In [79], the authors use stochastic models and recommend SP5, SP6 and SP7 with small values of λ for expensive simulation-based problems as is the case with the aircraft range problem. Accordingly, those three formulations were tested with kriging models and $\lambda = 0.01$. On this particular problem, $\text{SP5}_{0.01}$ turns out to be the best formulation. Consequently, extended aggregate models were compared to kriging models on $\text{SP5}_{0.01}$. Figures 4.7a and 4.7b show the data profiles of $\text{SP5}_{0.01}$ with smooth and nonsmooth alternatives, and kriging models. The latter perform indeed better than the extended aggregate models on $\text{SP5}_{0.01}$ at tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, meaning that the best formulations with real stochastic models are not necessarily the same than the best ones with the proposed extended aggregate models.

As a result, the fair comparison is not between kriging models and extended aggregate models *within the same formulation*, but rather between each type of models coupled with its best formulation, that is $\text{SP3}_{0.1}$ with the smooth alternative, SP8 with the nonsmooth alternative, and $\text{SP5}_{0.01}$ with kriging models. Figures 4.7c and 4.7d show the data profiles of the above combinations and in addition quad search as a reference. At tolerance $\tau = 10^{-7}$, the performances are alike but at tolerance $\tau = 10^{-9}$, the extended aggregate models coupled with the suitable formulations solve more problems, faster than kriging models with their own appropriate formulation, and faster than quad search. To tolerance $\tau = 10^{-9}$, the smooth and nonsmooth alternatives solve 87.5% and 75% of the problems, respectively, while kriging models and quad search solve 72.5% and 57.5% of the problems, respectively.

Since the different approaches require some non negligible amount of internal computation, it is appropriate to compare them not only in terms of the number of evaluations, but also in terms of the total real optimization time. It was highlighted in Section 4.4.1 that kriging

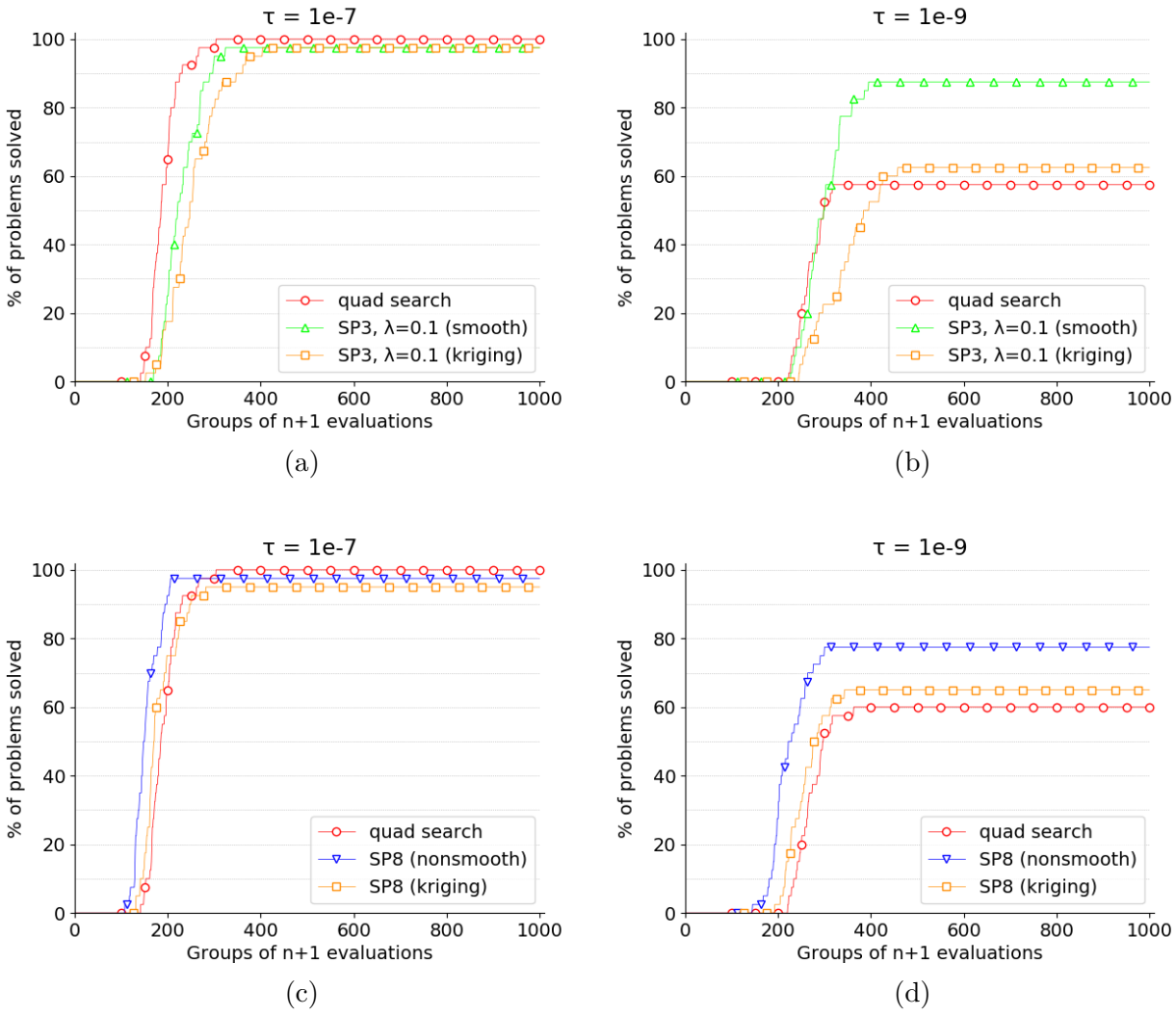


Figure 4.6 Data profiles. Figures 4.6a and 4.6b show quad search vs. $SP3_{0.1}$ with smooth uncertainty vs. $SP3_{0.1}$ with kriging models. Figures 4.6c and 4.6d show quad search vs. SP8 with nonsmooth uncertainty vs. SP8 with kriging models on the aircraft range problem.

models were prohibitively long to train when optimizing cheap analytical problems. The same question is addressed more thoroughly with the aircraft range problem. Figure 4.8 shows the *time data profiles* of formulation $SP3_{0.1}$ with the smooth alternative, formulation SP8 with the nonsmooth alternative, and formulation $SP5_{0.01}$ with kriging models. In a time data profile, the proportion of problems solved is not a function of the number of evaluations anymore but a function of the real computation time instead. The profiles suggest that even with a more expensive, real-world problem, kriging models are especially long to train. At tolerance $\tau = 10^{-9}$, $SP3_{0.1}$ with the smooth alternative solves 87.5% of the problems within 95 seconds and SP8 with the nonsmooth alternative solves 75% of the problems within 73

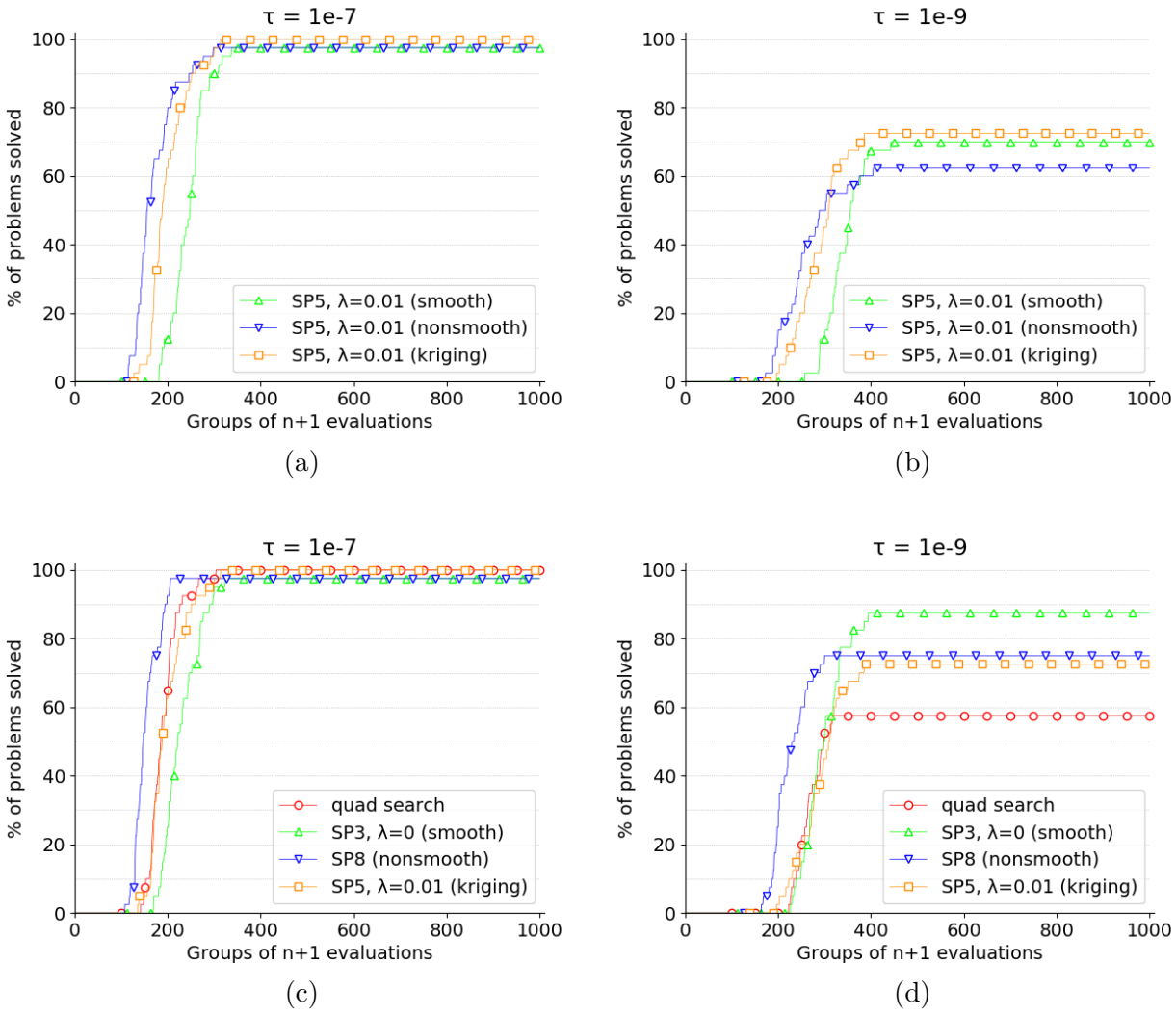


Figure 4.7 Data profiles. Figures 4.7a and 4.7b show $SP_{0.01}$ with smooth uncertainty vs. $SP5_{0.01}$ with nonsmooth uncertainty vs. $SP5_{0.01}$ with kriging models. Figures 4.7c and 4.7d show quad search vs. $SP3_{0.1}$ with smooth uncertainty vs. SP8 with nonsmooth uncertainty vs. $SP5_{0.01}$ with kriging models on the aircraft range problem.

seconds. After the same amount of time, $SP5_{0.01}$ with kriging models has solved less than 10% of the problems, and requires 728 seconds to solve 65% of the problems. Some instances even require more than 1500 seconds. Based on the present results, the proposed extended aggregate models constitute a cheaper and more efficient alternative to kriging models.

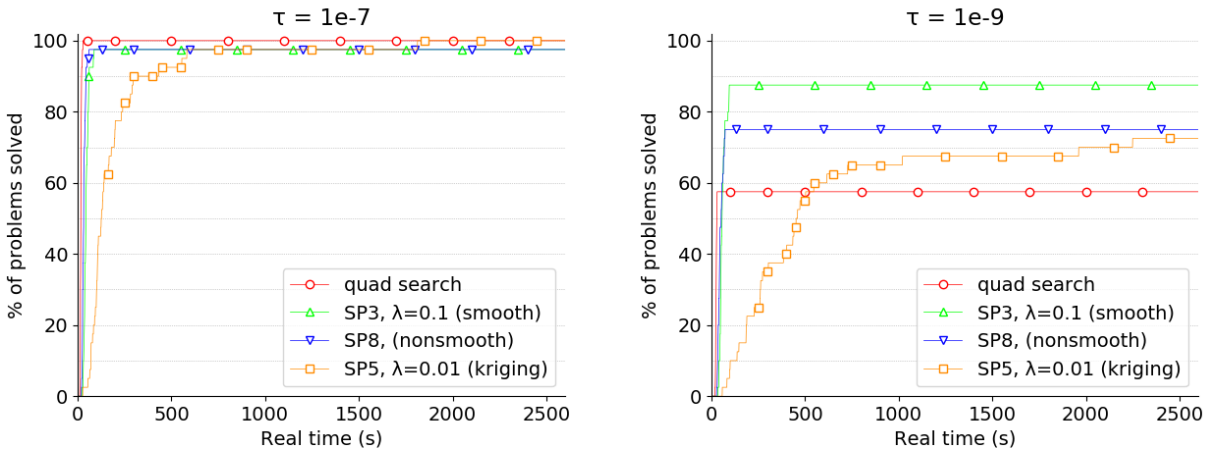


Figure 4.8 Time data profiles. quad search vs. $SP3_{0.1}$ with smooth uncertainty vs. SP8 with nonsmooth uncertainty vs. $SP5_{0.01}$ with kriging models on the aircraft range problem.

4.4.3 The simplified wing problem

The simplified wing problem [84] is also an MDO problem. It aims at minimizing the drag of a wing by taking into account aerodynamics and structure. This problem is smooth but has several local optima. It has $n = 7$ bounded variables and $m = 3$ constraints.

Ten starting point have been randomly sampled in the bounded space of variables with Latin hypercube sampling. The evaluation budget is $600(n + 1)$. Unlike the previous problems and for the rest of the study, the kriging models have not been tested because of their heavy computational cost, and not all the formulations have been tested but rather a subset of the most promising ones found on the aircraft range problem: $SP1_1$, $SP2_0$, $SP3_{0.1}$, $SP7_1$ and SP8 for the smooth alternative; and $SP2_{0.1}$, $SP3_{0.01}$, SP4, $SP6_{0.1}$ and SP8 for the nonsmooth alternative. Among the above formulations, the best ones found for this problem with the smooth and nonsmooth alternatives are SP8 and SP4, respectively. SP4 consists in maximizing $\mathrm{EFI}(x)$. Figure 4.9 shows the data profiles of no search, quad search, SP8 with smooth uncertainty, SP4 with nonsmooth uncertainty, and DFN at tolerances $\tau = 10^{-1}$ and $\tau = 10^{-3}$. Lower tolerances resulted in data profiles that were too flat, especially because of the high sensitivity to the seed of NOMAD on this problem.

On this MDO problem, the extended aggregated models perform better than no search, quad search and DFN. SP4 with the smooth uncertainty is especially good at tolerance $\tau = 10^{-3}$, solving 60% of problems while quad search only solves 20%. As with the aircraft range problem, the nonsmooth alternative solves problems faster than the smooth one. However, on the simplified wing problem the smooth alternative is by far the most efficient eventually.

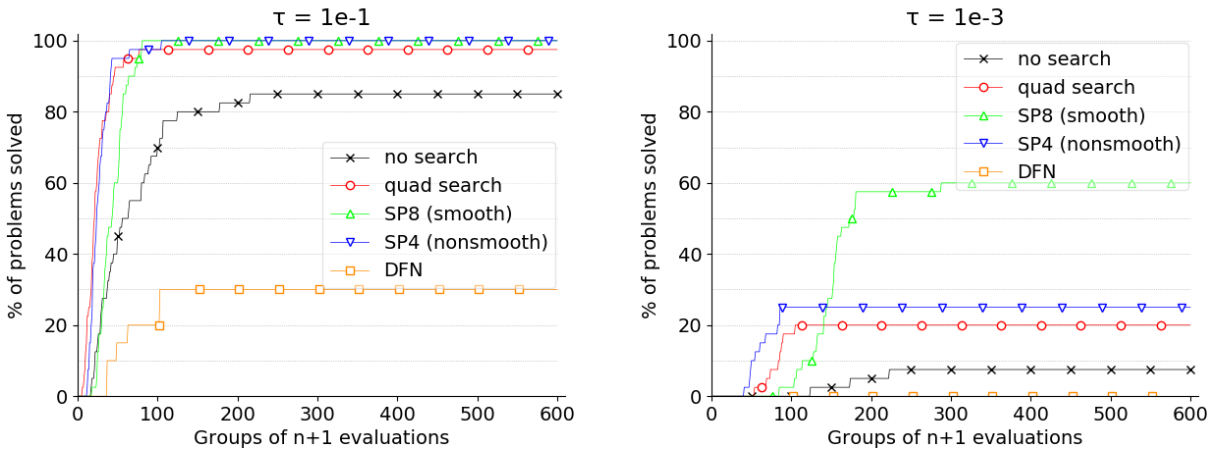


Figure 4.9 Data profiles. no search vs. quad search vs. SP8 with smooth uncertainty vs. SP4 with nonsmooth uncertainty vs. DFN on the simplified wing problem.

4.4.4 The solar1 problem

The solar1 problem is part of a set of nine concentrated solar power simulation problems [36] that serve as a benchmark for blackbox optimization solvers, available at `github.com/bbopt/solar`. The solar1 problem aims at maximizing the heliostat field energy output throughout one day under constraints of field geometry and cost. The problem is noisy and has several local optima. It has $n = 9$ variables, among which one is discrete, and $m = 5$ constraints. The solar1 problem has the specificity of having two additional adjustable parameters: the seed and the number of replications. Since the problem contains some stochasticity, the seed for the random generator can be chosen. It is *not* the same seed as the one of NOMAD. Two instances of solar1 with two different seeds are considered as two different problems in this work. The other parameter enables to replicate the evaluations in order to smooth the problem and compensate the noise. It was fixed to ten in the experiments.

In order to generate several problem instances, fifteen different seeds are chosen for the problem - not for NOMAD- instead of multiple starting points. Because the problem is especially expensive, NOMAD has been tested with only one seed, and not four, in order to reduce the number of optimization runs. The evaluation budget is $800(n + 1)$. As with the previous problem, only the best formulations from the aircraft range problem have been tested. The best one among them is SP8 for both smooth and nonsmooth alternatives. They manage to yield better results than no search, however, quad search is clearly the best algorithm on this problem.

Figure 4.10 shows the data profiles of SP8 with smooth and nonsmooth alternatives along with quad search. DFN and no search are not represent because they do not manage to solve one problem even at the largest tolerance. At tolerance $\tau = 10^{-1}$, the smooth alternative manages to solve as many problems as quad search. However, for higher tolerances, the latter is by far the best alternative. The poor performance of the extended aggregate models can be attributed to the stochasticity of the problem that importantly deteriorates the models. Nonetheless, it also impacts quad search. The superior performance of the latter is also due to the nature of the constraints: three out of five are linear, and one is cubic, which gives a non negligible advantage to quadratic models.

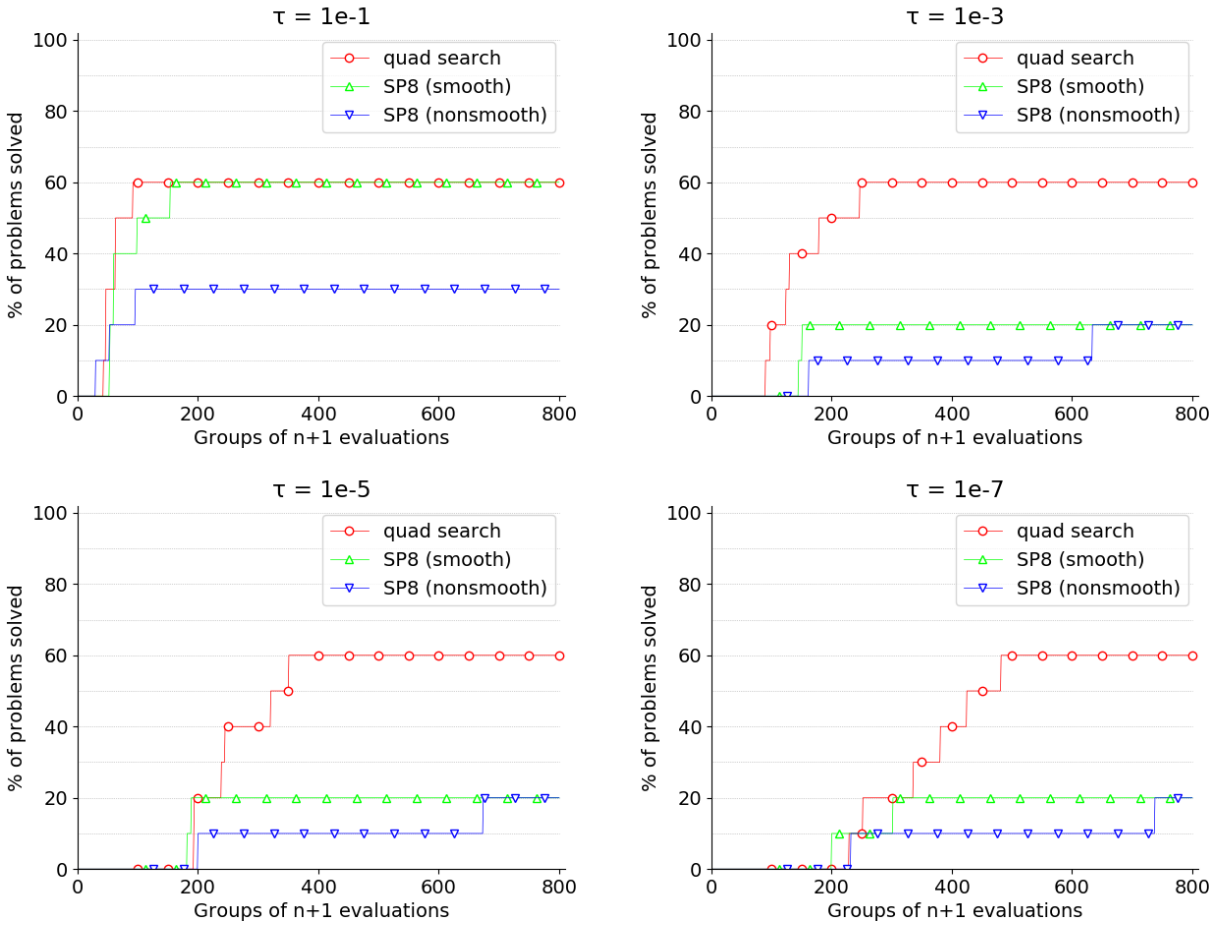


Figure 4.10 Data profiles. quad search vs. SP8 with smooth uncertainty vs. SP8 with nonsmooth uncertainty on solar1.

4.4.5 The styrene problem

The styrene problem is a chemical engineering simulator for styrene production described in [8] and available at `github.com/bbopt/styrene`. It aims at maximizing the net present value of the styrene production process under structural, chemical and financial constraints with variables comprised of physical parameters and structure. The problem is deterministic but nonsmooth and with omnipresent hidden constraints, i.e., the simulation often fails to return a value even when all constraints are met. A random sampling resulted in almost 60% of failures in [38]. In addition, four constraints are binary. The problem has $n = 8$ bounded variables and $m = 11$ constraints.

Feasible regions may by especially hard to find on this problem. Consequently, twelve starting points were generated in three relatively easy regions. The evaluation budget is $600(n + 1)$. Like the previous problem solar1, the same subset of the most promising formulations found on the aircraft range problem has been tested. The best formulations are SP1_1 with the smooth alternative and $\text{SP3}_{0.01}$ with the nonsmooth alternative.

Figure 4.11 shows the data profiles of no search, quad search, SP1_1 with the smooth uncertainty and $\text{SP3}_{0.01}$ with the nonsmooth uncertainty at tolerance $\tau = 10^{-1}$ and $\tau = 10^{-2}$. On this problem, the extended aggregate models perform better than both no search and quad search. Besides, unlike all the others problems the latter yields worse results than no search due to the binary constraints. DFN struggles to find feasible solutions and the results are not presented.

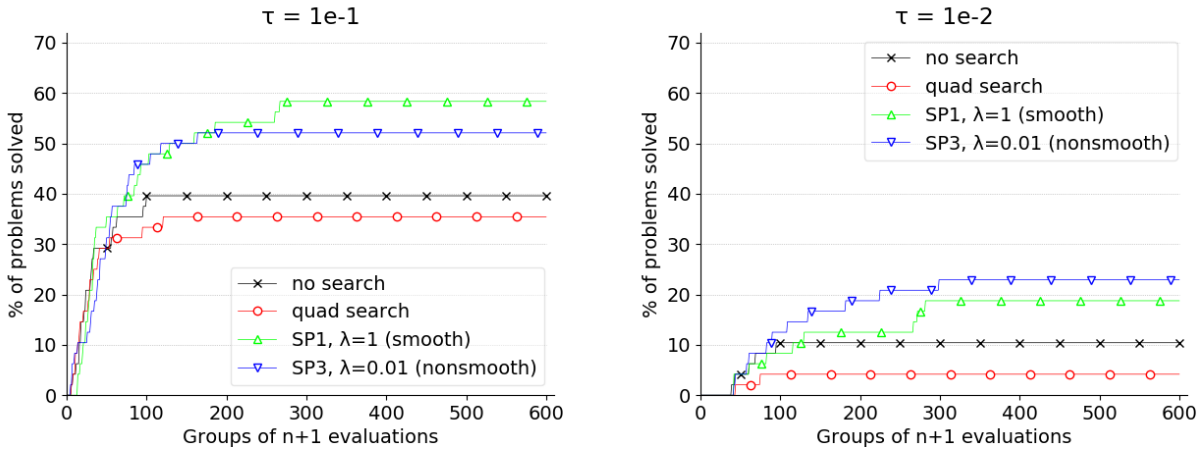


Figure 4.11 Data profiles. no search vs. quad search vs. SP1_1 with smooth uncertainty vs. $\text{SP3}_{0.01}$ with nonsmooth uncertainty on styrene.

4.4.6 Results of SHEBO

Unlike the other algorithms in this work, SHEBO does not take into account a starting point as an input, thereby making the comparison through data profiles impossible. In order to analyze the results, the best value found for a given optimization run is denoted by f^*, the real time (in minutes) needed to reach f^* is denoted by t^*, and the number of evaluations needed to reach f^* is denoted by k^*. Table 4.2 shows for most of the solvers seen before the median f^*, t^* and k^* on each problem considering all starting points and seeds, denoted by f^*_{m}, t^*_{m} and k^*_{m}, respectively, as well as the best f^* found on all runs, denoted by f^*_{best}, and the total real time needed for all runs, denoted by t_{tot}. For every problem, except the analytical problems, the three quantities are shown for no search, quad search, the best formulation found with the smooth uncertainty, the best formulation found with the nonsmooth uncertainty, DFN and SHEBO. However, SHEBO has been run only once on each problem due to its long running time. Consequently, all the values shown on the lines corresponding to SHEBO relate only to a single run, and f^*_{m} is equal to f^*_{best}.

For solar1 and styrene, SHEBO has a better f^*_{best} than the median f^* of all algorithms. However, on solar1 the proposed approach, both smooth and nonsmooth, manages to find a better f^*_{best} over all 10 runs after only 1.5 times more real time than the single run of SHEBO. On styrene, all the MADS algorithms manage to find a better f^*_{best} than SHEBO after all 48 runs, with significantly more real time though. On the aircraft range and the simplified wing problems, the proposed approach not only finds a better f^*_{best} than SHEBO but also a better f^*_{med} after only a fraction of the time per run. On the simplified wing problem, DFN also finds a better f^*_{best} than SHEBO with 1.7 times more real time. Regarding the number of evaluations, SHEBO requires more function evaluations to reach its best value than the other algorithms, except on the simplified wing problem. Overall, on the present experiments, one single run with SHEBO guaranties a decent value, only it demands a much larger computation time which is better invested running the proposed approach multiple times.

One final remark can be made about the real time required by the extended aggregate models compared to the quadratic models. The proposed approach is not always longer than quad search, both in terms of median time and total time. The quadratic models indeed tend to result in very long runs when they find a good bassin of solutions. Conversely, they result in very short runs when then they do not manage to find a good solution, depending on the starting point.

Table 4.2 Results of SHEBO compared to other algorithms.

	Aircraft range (40 runs except for SHEBO)					Simplified wing (40 runs except for SHEBO)				
	f^*_{m}	t^*_{m}	k^*_{m}	f^*_{best}	t_{tot}	f^*_{m}	t^*_{m}	k^*_{m}	f^*_{best}	t_{tot}
no search	-3964.204696	1	4140	-3964.204700	118	-16.4059	3	2405	-16.6119	397
quad search	-3964.204698	5	3064	-3964.204701	834	-16.5808	12	1291	-16.6120	2181
smooth	-3964.204699	10	3218	-3964.204701	1836	-16.6063	4	2115	-16.6120	921
nonsmooth	-3964.204699	9	2555	-3964.204700	1614	-16.5924	2	836	-16.6120	393
DFN	-1749.256329	0.1	73	-3143.101404	18	-15.0774	3	1134	-16.6112	588
SHEBO	·	784	4404	-3723.074357	3210	·	64	2190	-16.5501	356
	solar1 (10 runs except for SHEBO)					styrene (48 runs except for SHEBO)				
	f^*_{m}	t^*_{m}	k^*_{m}	f^*_{best}	t_{tot}	f^*_{m}	t^*_{m}	k^*_{m}	f^*_{best}	t_{tot}
no search	-508184.0	190	4064	-660888.3	2384	-29305150	12	1541	-33613200	3611
quad search	-723623.2	510	3795	-835124.4	6837	-29301200	41	1675	-33000800	9750
smooth	-677669.7	305	3688	-849192.5	6303	-32704300	49	1809	-33705600	19015
nonsmooth	-658837.2	224	3359	-805799.5	6190	-32235250	48	1647	-33697400	16086
DFN	-332582.8	68	1448	-391849.8	735	-22851550	0.3	38	-28517600	49
SHEBO	·	2865	6190	-815086.2	4475	·	741	5365	-32873400	774

4.5 Discussion

This work proposes an extension to ensembles of models that enables to compute an uncertainty at any given point. The resulting extended aggregate models behave like stochastic models, i.e., they produce at any given point x a prediction $\hat{f}(x)$ and also an uncertainty $\hat{\sigma}(x)$, thus enabling to use tools inspired by Bayesian optimization. The proposed extended aggregate models are incorporated into the search step of MADS where at each iteration a surrogate subproblem derived from Bayesian optimization is solved in order to come up with new candidate points. The proposed approach may be used in any direct search method based on the search-poll paradigm, or in any approach akin to efficient global optimization if adapted. Any ensemble of models can be used along with any weight attribution technique provided that at least two models have a strictly positive weight at any moment.

The resulting algorithm has been tested on seven analytical problems, two multidisciplinary optimization problems and two simulation problems. The results show that the proposed extended aggregate models incorporated into MADS find better solutions than MADS without search step or with the help of quadratic models on three expensive problems out of four. They also find better solutions than the stochastic models that they replace while requiring much less computational time. It should be noted that the models used to build the aggregate models must remain moderately expensive to compute, otherwise the method might loose its advantage in terms of computation time. The proposed approach does not show an advantage over quadratic models on analytical problems. In addition, the latter yield better results on the solar1 problem which most of the constraints are linear. The comparison to other solvers shows that the proposed approach has a clear advantage over DFN, and is more interesting than SHEBO when given the same computation time. An extended study of the various sub-problem formulations has not been conducted but based on the present results the formulations SP1, SP3 and SP8 can be recommended.

Future work may explore the uncertainty for the constraint independently from that of the objective, e.g., smooth uncertainty for the objective together with nonsmooth uncertainty for the constraint, since in this work the two were coupled. Other weight attribution techniques than that described in Section 4.3.2 may also be considered. The influence of the simplex used to build simplex gradients in (4.2) as well as the positive spanning set in (4.3) have not been studied in this work. Besides, the parameter λ of the surrogate subproblems has been carefully selected for each formulation but remains constant over the optimization once determined. It might instead be dynamically updated depending on the result of the search or merely follow a predetermined trend like decreasing with the number of iterations. The parameter α in (4.7) is proportional to the global variance of the cache. It could be refined

in order to represent local trends better, for instance by taking into account the values of the cache only in a restricted area around the evaluated point, or by removing outliers. Finally, the formulations were chosen on a purely empirical basis. Little effort has been made to finely understand the behaviour and the performance thereof. A thorough analysis of the benefits of each formulations in the context of extended aggregate models might be a judicious undertaking.

A Positive spanning set and simplex construction

The simplex and the positive spanning set described below are built in the *scaled* search space. Before constructing the models, the NOMAD software used in this work scales each input variable x_i, $i \in \{1, 2, \ldots, n\}$, using the mean and the variance of the points of the cache. This is done to give the same importance to all the variables regardless of their initial amplitude. Consequently, the simplex and the positive spanning set are isotropic in the scaled search space, but not in the actual search space.

— The simplex centred on $x \in \mathbb{R}^n$ used to build the simplex gradients $\nabla_S f(x)$ in Equation (4.2) is

$$\{x + 0.001 d_i \ : \ 1 \leq i \leq n+1\}\}$$

where the directions d_i, $i \in \{1, 2, \ldots, n+1\}$, are constructed according to the following procedure:

$$d_i = \begin{cases} e_i - \dfrac{1 + \frac{1}{\sqrt{n+1}}}{n} \times [1, 1, \ldots, 1]^\top, & \text{if } i \in \{1, 2, \ldots, n\} \\ \dfrac{1}{\sqrt{2(n+1)}} \times [1, 1, \ldots, 1]^\top, & \text{if } i = n+1 \end{cases}$$

where e_i is the vector $[0, \ldots, 1, \ldots, 0]^\top$ with value 1 at the ith position. This procedure forms a regular simplex centred on x with side length equal to $\sqrt{2}$ in any dimension. The factor 0.001 was chosen empirically on preliminary tests.

— The positive spanning set centred on $x \in \mathbb{R}^n$ used in Equation (4.3) is

$$\{x \pm 0.005 e_i \ : \ 1 \leq i \leq n\}$$

This positive spanning set contains $2n$ elements. The factor 0.005 was chosen empirically on preliminary tests.

B Surrogate subproblem formulations

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & \hat{f}(x) - \lambda\hat{\sigma}_f(x) \\ \text{s.t.} \quad & \hat{c}_j(x) - \lambda\hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \end{aligned} \tag{SP1-Fσ}$$

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & \hat{f}(x) - \lambda\hat{\sigma}_f(x) \\ \text{s.t.} \quad & \mathrm{P}(x) \geq p_c \end{aligned} \tag{SP2-FσP}$$

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & -\mathrm{EI}(x) - \lambda\hat{\sigma}_f(x) \\ \text{s.t.} \quad & \hat{c}_j(x) - \lambda\hat{\sigma}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \end{aligned} \tag{SP3-EIσ}$$

$$\min_{x\in\mathcal{X}} \quad -\mathrm{EFI}(x) \tag{SP4-EFI}$$

$$\min_{x\in\mathcal{X}} \quad -\mathrm{EFI}(x) - \lambda\hat{\sigma}_f(x) \tag{SP5-EFIσ}$$

$$\min_{x\in\mathcal{X}} \quad -\mathrm{EFI}(x) - \lambda\hat{\sigma}_f(x)\mu(x) \tag{SP6-EFIμ}$$

$$\min_{x\in\mathcal{X}} \quad -\mathrm{EFI}(x) - \lambda(\mathrm{EI}(x)\mu(x) + \mathrm{P}(x)\hat{\sigma}_f(x)) \tag{SP7-EFICμ}$$

$$\min_{x\in\mathcal{X}} \quad -\mathrm{PFI}(x) \tag{SP8-PFI}$$

CHAPITRE 5 DISCUSSION GÉNÉRALE

Dans ce mémoire, une extension aux ensembles de modèles est proposée afin que ceux-là puissent fournir, en plus d'une prédiction, une mesure d'incertitude en tout point de l'espace de recherche. Cette incertitude est fondée sur la corrélation entre les modèles, idée qui a été identifiée dans la littérature mais qui n'avait pas été exploitée à cette fin jusqu'à maintenant. La méthode ainsi conçue permet d'imiter des modèles stochastiques sans y avoir recours. Il serait toutefois possible que les ensembles de modèles incluent des modèles stochastiques ce qui permettrait d'utiliser l'incertitude que ceux-ci procurent, mais ce n'est pas l'objet de ce travail.

Afin d'éprouver les modèles agrégés augmentés d'une mesure de l'incertitude, ils ont été intégrés dans l'algorithme MADS, où ils servent à sélectionner des points candidats à l'étape de recherche globale. Néanmoins, ces modèles peuvent être utilisés en dehors de MADS, dans n'importe quelle méthode ayant habituellement recours à des modèles stochastiques. Les tests montrent que sur des problèmes difficiles, c'est-à-dire fortement non linéaires et parfois non lisses, les modèles proposés donnent de meilleurs résultats que les modèles quadratiques, ces derniers étant pourtant une référence en optimisation de boîtes noires et ayant fait l'objet de nombreux développements. Toutefois, la méthode proposée implique le choix d'une formulation du sous-problème substitut utilisé par MADS, ce qui, même au terme de ce mémoire, n'est pas une question résolue et devrait faire l'objet de futurs travaux. Les modèles quadratiques, quant à eux, ne nécessitent pas un tel choix.

La méthode a aussi été comparée, toujours par le biais de MADS, à deux autres algorithmes : DFN et SHEBO. Ces deux algorithmes n'abordent pas exactement les mêmes classes de problèmes : DFN est conçu pour des problèmes non lisses et SHEBO ne gère que les contraintes cachées en plus des contraintes linéaires, ce qui excluent les contraintes qui sont elles-mêmes des boîtes noires. Il pourrait être avancé que cela nuit à la légitimité des résultats, toutefois cela souligne aussi la rareté des algorithmes d'optimisation de boîtes noires qui permettent de gérer des contraintes générales et qui ont une implémentation disponible en ligne.

La contribution de ce mémoire réside essentiellement dans la conception d'un nouvel outil. Celui-ci exploite les disparités entre les prédictions des modèles d'une même fonction afin de quantifier l'incertitude que l'on peut avoir sur les valeurs de cette fonction. Les test numériques exposés sont une façon d'examiner ces modèles mais leur utilisation ne doit pas se limiter au contexte choisi dans ce mémoire.

CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS

6.1 Synthèse des travaux

Ce mémoire propose une extension aux ensembles de modèles leur permettant de fournir en tout point non seulement une prédiction mais aussi une mesure de l'incertitude. Celle-ci se décline en deux versions : une pour l'objectif et une pour les contraintes ; et en deux alternatives : une lisse et une non-lisse. La version pour l'objectif est une mesure des disparités entre les variations des modèles. En un point donné, plus les modèles sont décorrélés, plus l'incertitude est grande. La version pour les contraintes mesure quant à elle les disparités entre les signes des contraintes. En un point donné, l'incertitude augmente si les modèles des contraintes prédisent des signes différents. Les alternatives lisse et non-lisse se comportent de façon similaire mais dans le premier cas l'incertitude peut prendre en un point donné un ensemble continu de valeurs, alors que dans le second un ensemble fini seulement. Les modèles ainsi construits se comportent comme des modèles stochastiques, i.e. ils produisent en tout point x une prédiction $\hat{f}(x)$ et aussi une incertitude $\hat{\sigma}(x)$, ce qui permet d'utiliser des outils issus de l'optimisation bayésienne. Ces outils ont été adaptés pour pallier au manque d'information probabiliste, comme la fonction de répartition, propre aux modèles stochastiques. Les modèles proposés ont ensuite été intégrés dans l'algorithme MADS. À la phase de recherche globale, un sous-problème faisant appel aux outils de l'optimisation bayésienne est résolu afin de trouver de nouveaux points candidats. Plusieurs formulations de ce sous-problème ont été expérimentées afin d'identifier les meilleures.

L'algorithme mis au point fut éprouvé sur des divers problèmes : sept problèmes analytiques, deux problèmes d'optimisation multi-disciplinaire et deux problèmes de simulation. Les résultats montrent que sur la plupart des problèmes difficiles et coûteux, MADS muni des nouveaux modèles trouve de meilleures solutions que MADS sans recherche globale ou avec des modèles quadratiques. Sur les quelques problèmes sur lesquels MADS muni de modèles stochastiques fut testé, on constate que les nouveaux modèles agrégés sont plus efficaces tout en étant bien moins coûteux en temps de calcul. Toutefois, sur les problèmes analytiques, les modèles quadratiques s'avèrent plus efficaces que l'approche proposée tant en terme de précision que de temps de calcul. Sur un des problèmes de simulation, les modèles quadratiques sont également meilleurs à cause de la présence de contraintes linéaires. Par ailleurs, la comparaison avec les deux autres solveurs montre d'une part que DFN est très rapide mais ne trouve pas de solutions compétitives, et d'autre part que SHEBO permet de trouver des solutions parfois satisfaisantes mais en un temps très long. Sur la plupart des problèmes, il

est préférable d'exécuter plusieurs fois l'algorithme mis au point plutôt que d'exécuter une seule fois SHEBO.

Sur la base de ces résultats, on peut dire que les modèles agrégés munis d'une mesure d'incertitude intégrés dans MADS constituent une approche efficace, tant en terme de précision que de temps de calcul, sur des problèmes difficiles qui ne présentent que très peu de linéarité. Autrement, des versions de MADS plus classiques sont à recommander.

6.2 Limitations et améliorations futures

L'approche proposée permet d'utiliser une sélection arbitraire de modèles pourvu qu'il y en ait plus de deux. Toutefois, pour que la méthode ne perde pas son avantage en temps de calcul par rapport aux modèles stochastiques, il importe de ne pas choisir des modèles longs à construire ou mettre à jour. Par ailleurs, la méthode peut être jugée excessivement complexe et lourde en calculs si l'on souhaite aborder des problèmes relativement simples, c'est-à-dire analytiques ou présentant des comportements linéaires. Dans ce cas, de simples modèles quadratiques sont plus à même d'imiter les fonctions du problème tout en étant moins coûteux.

L'enjeu principal des travaux à venir est d'étudier en détails le comportement des diverses formulations du sous-problème substitut de la phase de recherche globale de MADS. Dans ce travail, les formulations ont été testées et retenues de façon empirique. Or elle sont conçues pour mettre en avant des qualités différentes dans les points candidats. Étudier leur comportement dans le cadre des modèles agrégés et non pas celui des modèles stochastiques permettrait de mieux cerner le potentiel de la méthode, et de savoir quelles formulations sont meilleures en fonction des propriétés du problème à traiter. Par ailleurs, de futurs travaux pourraient étudier les deux alternatives pour l'incertitude séparément, c'est-à-dire par exemple l'incertitude lisse pour l'objectif, et la non-lisse pour les contraintes. D'autres méthodes d'attribution des poids des modèles peuvent être essayées, ainsi que d'autres constructions de simplexes ou d'ensembles générateurs positifs que celles utilisées ici. Entre outre, le choix des ensembles de modèles pourrait inclure des modèles stochastiques, ce qui n'est pas le cas dans ce travail, afin d'exploiter l'incertitude qu'ils possèdent naturellement. De plus, le facteur α permettant de mettre à l'échelle l'incertitude pourrait faire l'objet d'une réflexion plus élaborée, par exemple en ne prenant que les valeurs de la cache de façon locale et non pas globale. Enfin, le paramètre λ des formulations demeure constant une fois sélectionné. Comme suggéré dans [79], ce paramètre pourrait plutôt évoluer de façon dynamique au cours de l'optimisation.

RÉFÉRENCES

[1] M.A. Abramson, C. Audet, G. Couture, J.E. Dennis, Jr., S. Le Digabel, V. Rochon Montplaisir et C. Tribes : The NOMAD project. Logiciel disponible sur `https://www.gerad.ca/nomad`, 2021.

[2] M.A. Abramson, C. Audet, J.E. Dennis, Jr. et S. Le Digabel : OrthoMADS : A Deterministic MADS Instance with Orthogonal Directions. *SIAM Journal on Optimization*, 20(2):948–966, 2009.

[3] E. Acar et M. Rais-Rohani : Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37(3):279–294, 2009.

[4] S. Alarie, C. Audet, A.E. Gheribi, M. Kokkolaras et S. Le Digabel : Two decades of blackbox optimization applications. Rapport technique G-2020-58, Les cahiers du GERAD, 2020.

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

[6] C. Audet : A survey on direct search methods for blackbox optimization and their applications. *In* P.M. Pardalos et T.M. Rassias, éditeurs : *Mathematics without boundaries : Surveys in interdisciplinary research*, chapitre 2, pages 31–56. Springer, New York, NY, 2014.

[7] C. Audet, V. Béchard et J. Chaouki : Spent potliner treatment process optimization using a MADS algorithm. *Optimization and Engineering*, 9(2):143–160, 2008.

[8] C. Audet, V. Béchard et S. Le Digabel : Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. *Journal of Global Optimization*, 41(2):299–318, 2008.

[9] C. Audet et J. Côté-Massicotte : Dynamic improvements of static surrogates in direct search optimization. *Optimization Letters*, 13(6):1433–1447, 2019.

[10] C. Audet et J.E. Dennis, Jr. : Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[11] C. Audet et J.E. Dennis, Jr. : A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, 20(1):445–472, 2009.

[12] C. Audet, J.E. Dennis, Jr. et S. Le Digabel : Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm. *SIAM Journal on Optimization*, 19(3):1150–1170, 2008.

[13] C. Audet, J.E. Dennis, Jr. et S. Le Digabel : Trade-off studies in blackbox optimization. *Optimization Methods and Software*, 27(4–5):613–624, 2012.

[14] C. Audet et W. Hare : *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017.

[15] C. Audet et W. Hare : Model-based methods in derivative-free nonsmooth optimization. *In* A.M. Bagirov, M. Gaudioso, N. Karmitsa, M.M. Mäkelä et S. Taheri, éditeurs : *Numerical nonsmooth optimization*, chapitre 15. Springer, 2020.

[16] C. Audet, M. Kokkolaras, S. Le Digabel et B. Talgorn : Order-based error for managing ensembles of surrogates in mesh adaptive direct search. *Journal of Global Optimization*, 70(3):645–675, 2018.

[17] C. Audet, S. Le Digabel, V. Rochon Montplaisir et C. Tribes : NOMAD version 4 : Nonlinear optimization with the MADS algorithm. Rapport technique G-2021-23, Les cahiers du GERAD, 2021.

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

[19] M. Björkman et K. Holmström : Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering*, 1:373–397, 2000.

[20] A.J. Booker : Well-conditioned Kriging models for optimization of computer simulations. Rapport technique MandCT-TECH-00-002, Boeing Computer Services, Research and Technology, M/S 7L–68, Seattle, Washington 98124, 2000.

[21] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon et M.W. Trosset : A Rigorous Framework for Optimization of Expensive Functions by Surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.

[22] G.E.P. Box : Evolutionary operation : A method for increasing industrial productivity. *Appl. Statist.*, 6:81–101, 1957.

[23] L. Chen, H. Qiu, C. Jiang, X. Cai et L. Gao : Ensemble of surrogates with hybrid method using global and local measures for engineering design. *Structural and Multidisciplinary Optimization*, 57(4):1711–1729, 2018.

[24] T.D. Choi, O.J. Eslinger, C.T. Kelley, J.W. David et M. Etheridge : Optimization of automotive valve train components with implicit filtering. *Optimization and Engineering*, 1(1):9–27, 2000.

[25] F.H. Clarke : *Optimization and Nonsmooth Analysis*. John Wiley and Sons, New York, 1983. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

[26] A.R. Conn et S. Le Digabel : Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. *Optimization Methods and Software*, 28(1):139–158, 2013.

[27] A.R. Conn, K. Scheinberg et L.N. Vicente : *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[28] A.R. Conn et Ph.L. Toint : An algorithm using quadratic interpolation for unconstrained derivative free optimization. *In* G. Di Pillo et F. Gianessi, éditeurs : *Nonlinear Optimization and Applications*, pages 27–47. Plenum Publishing, New York, 1996.

[29] A.L. Custódio, H. Rocha et L.N. Vicente : Incorporating minimum Frobenius norm models in direct search. *Computational Optimization and Applications*, 46(2):265–278, 2010.

[30] C. Davis : Theory of positive linear dependence. *American Journal of Mathematics*, 76:733–746, 1954.

[31] J.E. Dennis, Jr. et V. Torczon : Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.

[32] G. Fasano, G. Liuzzi, S. Lucidi et F. Rinaldi : A Linesearch-Based Derivative-Free Approach for Nonsmooth Constrained Optimization. *SIAM Journal on Optimization*, 24(3):959–992, 2014.

[33] E. Fermi et N. Metropolis : Numerical solution of a minimum problem. Los Alamos Unclassified Report LA–1492, Los Alamos National Laboratory, Los Alamos, USA, 1952.

[34] R. Fletcher et S. Leyffer : Nonlinear programming without a penalty function. *Mathematical Programming*, Series A, 91:239–269, 2002.

[35] R. Fletcher, S. Leyffer et Ph.L. Toint : On the global convergence of a filter–SQP algorithm. *SIAM Journal on Optimization*, 13(1):44–59, 2002.

[36] M. Lemyre Garneau : Modelling of a solar thermal power plant for benchmarking black-box optimization solvers. Mémoire de D.E.A., Polytechnique Montréal, 2015. Available at `https://publications.polymtl.ca/1996/`.

[37] T. Goel, R.T. Haftka, W. Shyy et N.V. Queipo : Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.

[38] R.B. Gramacy et S. Le Digabel : The mesh adaptive direct search algorithm with treed Gaussian process surrogates. *Pacific Journal of Optimization*, 11(3):419–447, 2015.

[39] R.B. Gramacy et M.A. Taddy : dynaTree : An R Package Implementing Dynamic Trees for Learning and Design. Software available at `http://CRAN.R-project.org/package=dynaTree`, 2010.

[40] S. Greenhill, S. Rana, S. Gupta, P. Vellanki et S. Venkatesh : Bayesian Optimization for Adaptive Experimental Design : A Review. *IEEE Access*, 8:13937–13948, 2020.

[41] W. Hock et K. Schittkowski : *Test Examples for Nonlinear Programming Codes*, volume 187 de *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, Germany, 1981.

[42] R. Hooke et T.A. Jeeves : "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the Association for Computing Machinery*, 8(2):212–229, 1961.

[43] J. Jahn : *Introduction to the Theory of Nonlinear Optimization*. Springer, Berlin, 1994.

[44] D.R. Jones : A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[45] D.R Jones, M. Schonlau et W.J. Welch : Efficient Global Optimization of Expensive Black Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[46] C.T. Kelley : Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, 10:43–55, 1999.

[47] S. Kirkpatrick, C.D. Gelatt Jr. et M.P. Vecchi : Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[48] S. Kitayama, M. Arakawa et K. Yamazaki : Sequential approximate optimization using radial basis function network for engineering optimization. *Optimization and Engineering*, 12(4):535–557, 2011.

[49] S. Kodiyalam : Multidisciplinary aerospace systems optimization. Rapport technique NASA/CR-2001-211053, Lockheed Martin Space Systems Company, Computational AeroSciences Project, Sunnyvale, CA, 2001.

[50] J. Larson, M. Menickelly et S.M. Wild : Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[51] S. Le Digabel et S.M. Wild : A Taxonomy of Constraints in Simulation-Based Optimization. Rapport technique G-2015-57, Les cahiers du GERAD, 2015.

[52] S. Le Digabel et S.M. Wild : Web Page for "A Taxonomy of Constraints in Black-Box Simulation-Based Optimization". `http://www.mcs.anl.gov/~wild/taxcon`, 2016.

[53] L. Lukšan et J. Vlček : Test problems for nonsmooth unconstrained and linearly constrained optimization. Rapport technique V-798, ICS AS CR, 2000.

[54] A.L. Marsden, M. Wang, J.E. Dennis, Jr. et P. Moin : Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, 572:13–36, 2007.

[55] M.D. McKay, R.J. Beckman et W.J. Conover : A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[56] K.I.M. McKinnon : Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[57] N. Mladenović et P. Hansen : Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

[58] J. Mockus, V. Tiesis et A. Zilinskas : *The application of Bayesian methods for seeking the extremum*, pages 117–129. North-Holand, 1978.

[59] J.J. Moré et S.M. Wild : Benchmarking Derivative-Free Optimization Algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[60] J. Müller : An algorithmic framework for the optimization of computationally expensive bi-fidelity black-box problems. *INFOR : Information Systems and Operational Research*, 58(2):264–289, 2020.

[61] J. Müller et M. Day : Surrogate Optimization of Computationally Expensive Black-Box Problems with Hidden Constraints. *INFORMS Journal on Computing*, 31(4):689–702, 2019.

[62] J. Müller, J. Park, R. Sahu, C. Varadharajan, B. Arora, B. Faybishenko et D. Agarwal : Surrogate optimization of deep neural networks for groundwater predictions. *Journal of Global Optimization*, 2020.

[63] J. Müller et R. Piché : Mixture surrogate models based on Dempster-Shafer theory for global optimization problems. *Journal of Global Optimization*, 51(1):79–104, 2011.

[64] J.A. Nelder et R. Mead : A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[65] J.A. Nelder et R.W.M. Wedderburn : Generalized Linear Models. *Journal of the Royal Statistical Society : Series A (General)*, 135:370–384, 1972.

[66] N. Peremezhney, E. Hines, A. Lapkin et C. Connaughton : Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. *Engineering Optimization*, 46(11):1593–1607, 2014.

[67] C.E. Rasmussen et C.K.I. Williams : *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[68] R.G. Regis : Stochastic Radial Basis Function Algorithms for Large-scale Optimization Involving Expensive Black-box Objective and Constraint Functions. *Computers and Operations Research*, 38(5):837–853, 2011.

[69] R.G. Regis : Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243, 2014.

[70] R.G. Regis et C.A. Shoemaker : Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31:153–171, 2005.

[71] X. Ruan, P. Jiang, Q. Zhou, J. Hu et L. Shu : Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. *Structural and Multidisciplinary Optimization*, 62:3021–3052, 2020.

[72] J. Sacks, W.J. Welch, T.J. Mitchell et H.P. Wynn : Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[73] L. Sarrazin-McCann : Opportunisme et ordonnancement en optimisation sans dérivées. Mémoire de D.E.A., Polytechnique Montréal, 2018.

[74] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams et N. de Freitas : Taking the Human Out of the Loop : A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[75] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams et N. De Freitas : Taking the human out of the loop : A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[76] N. Srinivas, A. Krause, S.M. Kakade et M.W. Seeger : Gaussian Process Optimization in the Bandit Setting : No Regret and Experimental Design. Rapport technique 0912.3995, arXiv, 2010.

[77] M.A. Taddy, R.B. Gramacy et N.G. Polson : Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493):109–123, 2011.

[78] B. Talgorn, C. Audet, M. Kokkolaras et S. Le Digabel : Locally weighted regression models for surrogate-assisted design optimization. *Optimization and Engineering*, 19(1):213–238, 2018.

[79] B. Talgorn, S. Le Digabel et M. Kokkolaras : Statistical Surrogate Formulations for Simulation-Based Design Optimization. *Journal of Mechanical Design*, 137(2):021405–1–021405–18, 2015.

[80] H.A. Le Thi, A.I.F. Vaz et L.N. Vicente : Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimization. *TOP*, 20(1):190–214, 2012.

[81] D.J.J. Toal : Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. *Structural and Multidisciplinary Optimization*, 51(6): 1223–1245, 2015.

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

[83] R. Tournemenne, J.-F. Petiot, B. Talgorn, M. Kokkolaras et J. Gilbert : Brass Instruments Design Using Physics-Based Sound Simulation Models and Surrogate-Assisted Derivative-Free Optimization. *Journal of Mechanical Design*, 139(4):041401–01–041401–9, 2017.

[84] C. Tribes, J.-F. Dubé et J.-Y. Trépanier : Decomposition of multidisciplinary optimization problems : formulations and application to a simplified wing design. *Engineering Optimization*, 37(8):775–796, 2005.

[85] B. Van Dyke et T.J. Asaki : Using QR Decomposition to Obtain a New Instance of Mesh Adaptive Direct Search with Uniformly Distributed Polling Directions. *Journal of Optimization Theory and Applications*, 159(3):805–821, 2013.

[86] A. Verdério et E. W. Karas : On the construction of quadratic models for derivative-free trust-region algorithms. *EURO Journal on Computational Optimization*, 5(4):501–527, 2017.

[87] F.A.C. Viana, R.T. Haftka, S. Valder, Jr., S. Butkewitsch et M.F. Leal : Ensemble of Surrogates : a Framework based on Minimization of the Mean Integrated Square Error. *In 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials*, Schaumburg, IL, 2008.

[88] F.A.C. Viana, R.T. Haftka et L.T. Watson : Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, 2013.

[89] K.-K. Vu, C. D'Ambrosio, Y. Hamadi et L. Liberti : Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3):393–424, 2017.

[90] K.K. Vu, C. D'Ambrosio, Y. Hamadi et L. Liberti : Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3):393–424, 2017.

[91] S.M. Wild, R.G. Regis et C.A. Shoemaker : ORBIT : Optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

[92] S.M. Wild et C.A. Shoemaker : Global convergence of radial basis function trust region derivative-free algorithms. *SIAM J. Optimization*, 21(3):761–781, 2011.

[93] P. Ye, G. Pan et Z. Dong : Ensemble of surrogate based global optimization methods using hierarchical design space reduction. *Structural and Multidisciplinary Optimization*, 58:537–554, 2018.