Quelques contribtuions à l'optimisation globale

Cedric Malherbe

NNT: 2017SACLN052

THESE DE DOCTORAT
DE
L'UNIVERSITE PARIS-SACLAY
PREPAREE A
L'ECOLE NORMALE SUPÉRIEURE PARIS-SACLAY

ECOLE DOCTORALE N° 574
Mathématiques Hadamard

Spécialité de doctorat: Mathématiques appliquées

Par

Cédric Malherbe

Quelques contributions à l'optimisation globale

Thèse présentée et soutenue à Cachan, le 24 novembre 2017:

Composition du Jury:

M. Gilles Blanchard	Professeur, Université de Potsdam	Rapporteur
M. Jean-Philippe Vert	Directeur de Recherche, Mines ParisTech	Rapporteur
M. Remi Munos	Directeur de Recherche, Inria Lille	Examinateur
M. Olivier Teytaud	Chargé de Recherche, Inria Saclay	Examinateur
M. Alexandre Tsybakov	Professeur, ENSAE ParisTech	Examinateur
M. Vayatis Nicolas	Professeur, ENS Paris-Saclay	Directeur de thèse

Contents

Abstract

This work addresses the sequential optimization of an unknown and potentially non-convex function over a continuous and bounded set. The setup we consider assumes that the function evaluations are noiseless and that the derivative information of the function is unavailable. These problems are of particular interest when the function values are typically expensive to compute, and its derivatives are not available either because the evaluations result from some physical measures, or, more commonly, because it is the result of a possible heavy computer simulation. This includes in particular the problem of tuning the hyperparameters of large neural nets. In the first part, we consider the problem of designing sequential strategies which lead to efficient optimization of an unknown function under the only assumption that it has finite Lipschitz constant. We introduce and analyze a first algorithm, called LIPO, which assumes the Lipschitz constant to be known. Consistency and minimax rates for LIPO are proved, as well as fast rates under an additional Hölder like condition. An adaptive version of LIPO is also introduced for the more realistic setup where the Lipschitz constant is unknown and has to be estimated along with the optimization, and similar theoretical guarantees are shown to hold for the adaptive version of the algorithm. In the second part, we propose to explore concepts from ranking theory based on overlaying level sets in order to develop optimization methods that do not rely on the smoothness of the function. We start by observing that the optimization of the function essentially relies on learning the bipartite rule it induces. Based on this idea, we relate global optimization to bipartite ranking which allows to address the case of functions with weak regularity properties. Novel meta-algorithms for global optimization which rely on the choice of any bipartite ranking method are introduced and theoretical properties are provided in terms of statistical consistency and finite-time convergence toward the optimum. Eventually, the algorithms developed in the thesis are compared to existing state-of-the-art methods over typical benchmark problems for global optimization.

Notations

General

$\mathcal{X}$	A compact and convex subset of $\mathbb{R}^d$	p. 10
$\text{diam}(\mathcal{X})$	Diameter of $\mathcal{X}$	p. 51
$\text{rad}(\mathcal{X})$	Inner radius of $\mathcal{X}$	p. 51
$\|\cdot\|_2$	Standard ℓ_2-norm	p. 51
$B(x,r)$	ℓ_2-ball of radius r centered in x	p. 51
$\|X\|$	Number of elements of X	p. 11
$\vec{0}$	Zero vector of $\mathbb{R}^d$	p. 92

Optimization

f	The unknown function $\mathcal{X} \mapsto \mathbb{R}$ to be optimized	p. 10
d	Dimensionnality of the input space $\mathcal{X}$	p. 10
n	Number of function evaluations	p. 10
A	A global optimization algorithm	p. 51
$\text{Lip}(k)$	class of k-Lipschitz functions	p. 51
$\bigcup_{k\geq 0}\text{Lip}(k)$	class of Lipschitz functions	p. 51

Ranking

r_f	Ranking rule $\mathcal{X} \times \mathcal{X} \mapsto \{-1,0,1\}$ induced by f	p. 81
$L(r)$	Ranking loss	p. 84
$L_n(\cdot)$	Empricial ranking loss	p. 84
$\mathcal{R}$	A class of ranking rules	p. 82
$R_n(\mathcal{R})$	Rademacher complexity of $\mathcal{R}$	p. 89

Distributions and functions

$\mathbb{I}\{\cdot\}$	Indicator function taking values in $\{0,1\}$	p. 51
$\text{sgn}(\cdot)$	Sign function taking values in $\{-1,0,1\}$	p. 51
$\mu(\cdot)$	Lebesgue measure	p. 51
$\mathcal{B}(p)$	Bernoulli distribution of parameter p	p. 51
$\mathcal{U}(\mathcal{X})$	Uniform distribution over $\mathcal{X}$	p. 51

Introduction

I Présentation générale

Ce travail de thèse s'inscrit dans le domaine de l'optimisation et s'intéresse plus précisément au problème d'optimisation globale d'une fonction inconnue sur un ensemble continu et borné. L'optimisation apparaît dans de nombreux domaines et joue un rôle central dans beaucoup d'applications. Elle vise à trouver les paramètres d'entrée d'un système qui en optimisent la sortie ; l'objectif étant typiquement la maximisation d'une récompense ou la minimisation d'un coût. Cette branche des mathématiques est généralement découpée en sous domaines définis suivant la forme de la fonction que l'on cherche à optimiser et celle des contraintes. A titre d'exemple, de nombreux travaux sont entièrement dédiés à l'optimisation de fonctions linéaires, quadratiques ou convexes, et de nombreuses méthodes permettant de résoudre ce type de problème ont été proposées. Nous nous intéressons ici aux problèmes d'optimisation dont les valeurs de la fonction cible sont uniquement accessibles via des boîtes noires, souvent référencés dans la littérature comme globaux, non linéaires ou encore sans gradient. Dans ce cadre-là, la fonction qui relie l'entrée à la sortie n'est pas connue, mais l'on dispose tout de même d'une manière d'évaluer la sortie pour toute entrée de l'espace de recherche. Ces problèmes apparaissent notamment dans la conception de systèmes complexes, lorsque l'on cherche à optimiser le résultat de simulations numériques ou plus simplement lorsque la fonction que l'on souhaite optimiser est explicite mais ne présente aucune forme évidente de régularité comme la linéarité ou la convexité. Ainsi, les domaines de l'optimisation globale et de l'apprentissage automatique présentent au moins deux niveaux d'interactions. Premièrement, la mise en place de certains algorithmes d'optimisation repose sur l'utilisation de techniques issues de l'apprentissage comme le partitionnement de données ou encore l'ordonnancement, et inversement, certains algorithmes d'optimisation globale sont utilisés dans la pratique pour résoudre des problèmes de minimisation du risque empirique ou encore de calibration d'hyperparamètres. Motivés par ces applications, nous abordons dans ce travail le cas difficile où les évaluations peuvent être coûteuses en temps de calcul (comme en apprentissage profond), ce qui exige de concevoir une sélection séquentielle des entrées à évaluer qui utilise les mesures précédemment acquises. Comparativement aux problèmes d'optimisation plus classiques, la difficulté principale de ces problèmes vient du simple fait que, structurellement, aucune hypothèse portant sur la structure de la fonction que l'on souhaite optimiser n'est a priori exploitable. En effet, le système que l'on cherche à optimiser dans ces cas là peut présenter plusieurs optima locaux et les méthodes classiques qui visent à estimer un optimum local ne garantissent en aucun cas l'identification d'un optimum global. Pour tenter de répondre à ce problème, de nombreuses approches

basées sur diverses heuristiques ont été explorées. On pourra notamment citer les algorithmes génétiques, les méthodes bayésiennes, ou encore les techniques de partitionnement. Cependant, un des inconvénients majeurs de ces méthodes est qu'il est généralement difficile d'analyser ou de prédire leur comportement autrement que de façon empirique. Ainsi, leurs propriétés de convergence restent relativement mystérieuses et assez peu de travaux portant sur ces aspects sont référencés dans la littérature. La motivation principale de cette recherche est de parvenir à répondre à ces questions: (i) quel type de fonction peut-on espérer optimiser ? et (ii) comment peut-on définir la complexité d'un problème d'optimisation globale ? Pour tenter de répondre à ces questions, nous introduisons deux nouvelles stratégies d'optimisation globale, puis, en adaptant des concepts issus de la théorie de l'ordonnancement binaire et de l'apprentissage actif, nous identifions certaines classes de fonctions nous permettant d'analyser leurs propriétés de convergence. L'objectif étant d'apporter de nouveaux concepts visant à décrire l'efficacité de procédures d'optimisation globale par rapport à des notions génériques relatives à la complexité du problème.

II Énoncé du problème et méthodologie

Avant de présenter plus en détail le contenu du document, nous présentons le problème d'optimisation globale que nous considérons et discutons un certain nombre d'hypothèses méthodologiques que nous avons faites.

Énoncé du problème. Soit $\mathcal{X} \subset \mathbb{R}^d$ un sous-ensemble compact et convexe de l'espace euclidien d'intérieur non-vide et $f : \mathcal{X} \to \mathbb{R}$ une fonction inconnue, uniquement supposée atteindre un maximum sur son espace d'entrée $\mathcal{X}$. L'objectif en optimisation globale est d'identifier un point

$$x^\star \in \operatorname*{arg\,max}_{x \in \mathcal{X}} f(x)$$

tout en faisant un minimum d'évaluations de la fonction. Pour ce faire, une procédure d'optimisation séquentielle commence par évaluer la fonction $f(X_1)$ en un point initial et aléatoire $X_1 \in \mathcal{X}$, puis sélectionne à chaque étape $t \geq 1$, un nouveau point aléatoire $X_{t+1} \in \mathcal{X}$ qui dépend des évaluations précédentes $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ et est $\mathcal{F}_t$-mesurable où $\mathcal{F}_t = \sigma((X_1, f(X_1)), \ldots, (X_t, f(X_t)))$ correspond à la filtration générée par les évaluations précédentes, et reçoit la valeur de la fonction $f(X_{t+1})$ en ce point. Après un nombre $n \geq 1$ d'itérations, nous considérons qu'une procédure séquentielle renvoie un (des) point(s) ayant enregistré la plus haute évaluation de la fonction:

$$X_{\hat{\imath}_n} \quad \text{où} \quad \hat{\imath}_n \in \operatorname*{arg\,max}_{i=1\ldots n} f(X_i).$$

L'analyse fournie dans ce document considère par ailleurs que le nombre n d'évaluations n'est pas fixé et nous rappelons que les évaluations de la fonction sont supposées sans bruit. La Figure II.1 présente la structure d'une procédure d'optimisation séquentielle.

Exemples. Pour illustrer le type de problèmes auxquels on souhaite s'attaquer, nous présentons ici deux exemples d'applications qui nécessitent la résolution d'un problème d'optimisation global.

Entrée: $n \in \mathbb{N}^\star$, $\mathcal{X} \subset \mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialisation: Sélection de $X_1 \in \mathcal{X}$
Évaluation de $f(X_1)$, $t \leftarrow 1$

2. Itérations: Répéter tant que $t < n$:
Choix de $X_{t+1} \in \mathcal{X}$ en fonction des évaluations précédentes $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$
Évaluation de $f(X_{t+1})$, $t \leftarrow t + 1$

3. Sortie: Retour de $X_{\hat{\imath}_n}$ où $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Figure II.1: Squelette d'une procédure d'optimisation séquentielle

Exemple 1. (Design industriel). Ce problème, issu des travaux de Sarkar et al. (2016), est un exemple typique de problème d'optimisation boîte noire. Il consiste à identifier les coordonnées $(x_1^\star, y_1^\star), \ldots, (x_m^\star, y_m^\star)$ de $m = 40$ convertisseurs d'énergie de vagues positionnés dans une zone prédéfinie de l'océan qui maximisent le montant total d'énergie produite. Dans cet exemple, les valeurs de la fonction cible qui sont le montant d'énergie produite $f((x_1, y_1), \ldots, (x_m, y_m))$, sont estimées numériquement pour un unique jeu de coordonnées $(x_1, y_1), \ldots, (x_m, y_m)$ en résolvant un système complexe d'équations différentielles. Ainsi, ce problème peut être écrit comme suit:

$$(x_1^\star, y_1^\star), \ldots, (x_m^\star, y_m^\star) \in \underset{(x_i, y_i)_{i=1}^m \in [0,1]^{2m}}{\arg\max} f((x_1, y_1), \ldots, (x_m, y_m)),$$

où aucune information n'est a priori disponible sur la fonction cible et dont les évaluations nécessitent des simulations numériques coûteuses en temps de calcul, ce qui implique une exploration de l'espace de recherche avec peu d'évaluations.

Exemple 2. (Calibration de réseaux de neurones). Le second problème que nous présentons est un exemple qui consiste à identifier les valeurs de deux hyperparamètres d'un réseau de neurones. Plus précisément, pour une architecture fixée, une tâche classique consiste à identifier le seuil d'apprentissage $\eta^\star$ et le moment $\mu^\star$ d'une descente de gradient stochastique accélérée qui minimisent l'erreur empirique de classification calculée par validation croisée. Pour un jeu de données $(x_1, y_1), \ldots, (x_n, y_n)$ où $x_i \in \mathbb{R}^d$ et $y_i \in \{0, 1\}$, l'objectif est de résoudre le problème d'optimisation suivant:

$$(\ln(\eta^\star), \ln(1 - \mu^\star)) \in \underset{(\ln(\eta), \ln(1-\mu)) \in \mathcal{X}}{\arg\min} \frac{1}{5} \sum_{k=1}^{5} \frac{1}{|I_k|} \sum_{i \in I_k} \mathbb{I}\left\{ \hat{g}_{\eta,\mu}^k(x_i) \neq y_i \right\}$$

où

$$\hat{g}_{\eta,\mu}^k(x) = \mathbb{I}\left\{ \sigma_3(W_3^k \sigma_2(W_2^k \sigma_1(W_1^k x))) \geq 1/2 \right\}$$

sur $\mathcal{X} = [-5, 1/2] \times [-5, 0]$ où $I_1, \ldots, I_5$ sont cinq séries d'entiers formant une partition de $\{1, \ldots, n\}$, les matrices $W_1^k \in \mathbb{R}^{d \times 12}$, $W_2^k \in \mathbb{R}^{12 \times 8}$ et $W_3^k \in \mathbb{R}^{8 \times 1}$ sont calculées grâce à une descente de gradient stochastique accélérée de paramètres (η, μ) minimisant l'erreur

d'entropie croisée calculée sur le jeu de données $\{(x_i, y_i)\}_{i \notin I_k}$ et σ_1, σ_2 et σ_3 sont respectivement les fonctions d'activations: ReLu ($\max(x, 0)$), ReLu et sigmoid ($(1 + e^{-x})^{-1}$). Dans cet exemple, le temps de calcul est une fonction strictement croissante de la taille du réseau et du jeu de données et quasiment aucune information sur la structure de la fonction cible n'est exploitable.

Critères de performance. Les performances d'une procédure d'optimisation séquentielle sur une fonction $f : \mathcal{X} \to \mathbb{R}$ admettant un maximum global sur son espace d'entrée $\mathcal{X}$ sont mesurées, après $n \geq 1$ itérations, à travers la différence entre la valeur du maximum de la fonction et celle de son approximation:

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)$$

où $X_1, \dots, X_n$ est une séquence aléatoire de n points générés par l'algorithme sur la fonction f. Cependant, comme nous le verrons dans le Chapitre 3, il n'est pas toujours possible de borner ce critère lorsque la fonction cible n'est pas supposée être au moins localement régulière. Dans ce cas-là, en supposant que la fonction admette un unique optimiseur $x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$, les performances de l'algorithme seront alors mesurées à travers la distance euclidienne entre l'optimiseur et son approximation:

$$\left\| X_{\hat{\imath}_n} - x^\star \right\|_2 .$$

Il est à noter que lorsque la fonction objectif f est supposée admettre un unique optimiseur et être k-lipschitzienne pour un certain $k \geq 0$, le premier critère peut être contrôlé par le second à travers l'inégalité suivante: $\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \left\| X_{\hat{\imath}_n} - x^\star \right\|_2$. Nous pouvons désormais introduire la notion de convergence asymptotique d'une procédure séquentielle en optimisation globale.

Définition II.1. (Consistance d'une procédure séquentielle) *Un algorithme d'optimisation séquentiel est dit consistant sur un ensemble $\mathcal{F}$ de fonctions à valeurs réelles définies et admettant un maximum global sur l'espace d'entrée $\mathcal{X}$ si et seulement si*

$$\forall f \in \mathcal{F}, \quad \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x \in \mathcal{X}} f(x)$$

où $X_1, \dots, X_n$ est une séquence de $n \geq 1$ points aléatoires générés par l'algorithme sur la fonction f.

Ainsi, nous chercherons à déterminer sur quel ensemble $\mathcal{F}$ de fonctions un algorithme est consistant. Pour analyser les performances non-asymptotiques d'un algorithme, nous analyserons ses vitesses de convergence définies comme suit.

Définition II.2. (Vitesse de convergence) *Un algorithme d'optimisation converge à une vitesse $U_{n,\delta} \xrightarrow[n \to \infty]{} 0$, sur un ensemble $\mathcal{F}$ de fonctions définies et admettant un maximum sur $\mathcal{X}$ si et seulement si pour toute fonction $f \in \mathcal{F}$, tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0, 1)$, nous avons avec probabilité supérieure à $1 - \delta$:*

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq U_{n,\delta}$$

où $X_1, \dots, X_n$ est une séquence de points aléatoires générés par l'algorithme sur la fonction f.

La décroissance de la suite $\{U_{n,\delta}\}_{n\geq 1}$ décrit ici la vitesse de convergence d'un l'algorithme sur l'ensemble $\mathcal{F}$. On parlera notamment de vitesse de convergence polynomiale et on écrira $\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = O_{\mathbb{P}}(n^{-\alpha})$ lorsque pour tout $\delta \in (0,1)$, $U_{n,\delta} = O(n^{-\alpha})$ pour un certain $\alpha > 0$ ou encore de vitesse exponentielle lorsque $U_{n,\delta} = O(e^{-cn})$ pour une constante $c > 0$.

Discussion et méthodologie. Le problème présenté ci-dessus correspond à un certain nombre d'hypothèses méthodologiques. Nous commençons par présenter les hypothèses de fond qui caractérisent notre approche. Plus précisément, nous nous sommes restreints aux problèmes spécifiques d'optimisation où:

- l'espace de recherche est un sous-espace continu de l'espace euclidien. Nous nous focalisons donc sur des problèmes d'optimisation de paramètres continus, et ne considérons pas les problèmes d'optimisation combinatoires,

- aucune hypothèse portant sur la structure de la fonction que l'on souhaite optimiser n'est a priori exploitable. Toutefois, nous soulignons que des hypothèses de régularité comme la lipschitzianité, la quasi-continuité ou encore la régularité des ensembles de niveaux seront exploitées pour mettre en place des stratégies d'optimisation.

Ensuite, deux hypothèses correspondant plus à des commodités techniques que l'on peut éventuellement relâcher ont été formulées:

- les évaluations de la fonction sont supposées non bruitées,

- l'espace de recherche est un sous-espace compact et convexe de l'espace euclidien.

En effet, différentes pistes visant à étendre nos méthodes aux données bruitées ainsi qu'aux domaines non convexes ou non-bornés sont proposées dans les chapitres 1, 2 et 3, sections 1.1, 2.3 et 3.3. Finalement, deux hypothèses ne sont jamais explicitement spécifiées:

- aucune contrainte n'est fixée sur le nombre de paramètres à optimiser (*i.e.,* la dimension du problème),

- aucune hypothèse portant sur le temps de calcul nécessaire à l'évaluation de la fonction n'est formulée.

Il est cependant à noter que ces deux aspects ont un impact sur la complexité algorithmique et donc sur la vitesse de convergence observée. Ainsi, la mise en place de méthodes permettant d'optimiser des fonctions coûteuses en temps de calcul et ce pour de grandes dimensions reste un sujet de recherche actuel (voir, *e.g.*, les Exemples 1 et 2). Le but de notre approche est en effet de mettre en place des stratégies d'optimisation pour ce type de problèmes. La suite de ce chapitre contient un résumé des contributions de la thèse et des différents résultats théoriques connus en optimisation globale.

III Méthodes existantes

Afin de mettre en perspective les contributions de cette thèse, nous présentons ici une brève revue des principaux résultats théoriques connus en optimisation globale qui sont étroitement liés à notre travail.

Recherche purement aléatoire. Nous commençons par analyser la recherche purement aléatoire (Pure Random Search), qui consiste à évaluer la fonction sur une série de points $X_1, \dots, X_n$ indépendamment et uniformément distribués sur l'espace d'entrée $\mathcal{X}$. Du fait de la simplicité de la méthode, ses propriétés de convergence peuvent être analysées simplement en introduisant les ensembles de niveaux $\epsilon \geq 0$ de la fonction cible $\mathcal{X}_\epsilon := \{x \in \mathcal{X} : f(x) \geq \max_{x \in \mathcal{X}} f(x) - \epsilon\}$. Plus précisément, en notant $\mathcal{F}_{\text{lvl}} := \{f : \mathcal{X} \to \mathbb{R} \text{ t.q. } \forall \epsilon > 0, \ \ \mu(\mathcal{X}_\epsilon) > 0\}$ l'ensemble des fonctions dont tous les ensembles de niveaux sont de mesure non nulle où $\mu(\cdot)$ correspond à la mesure de Lebesgue, on peut directement obtenir la propriété de consistance de la stratégie:

$$\forall f \in \mathcal{F}_{\text{lvl}}, \ \ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x \in \mathcal{X}} f(x).$$

En supposant que la fonction inconnue f admette un optimiseur $x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$ unique tel qu'il existe $c_1, c_2 \geq 0$ et $\kappa_1 \geq \kappa_2 \geq 0$ pour lesquels nous ayons $c_1 \cdot \|x - x^\star\|_2^{\kappa_1} \leq f(x^\star) - f(x) \leq c_2 \cdot \|x - x^\star\|_2^{\kappa_2}$ pour tout $x \in \mathcal{X}$, on peut montrer que pour tout $\delta \in (0,1)$, nous avons avec probabilité au moins égale à $1 - \delta$:

$$C_1 \cdot \left(\frac{\delta}{2n}\right)^{\frac{\kappa_1}{d}} \leq \max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq C_2 \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{\kappa_2}{d}}$$

où $C_1 > 0$ et $C_2 > 0$ sont deux constantes numériques. Pour cette stratégie, les bornes obtenues sur les vitesses de convergence dans le cadre de fonctions localement régulières sont uniquement polynomiales en la dimensions $d \geq 1$ et donc sujettes au fléau de la dimension.

Méthodes de partitionnement. Les algorithmes DIRECT (Jones et al. (1993)) et SOO (Munos (2014)) utilisent un partitionnement de l'espace de recherche dans le but de séquentiellement évaluer la fonction et diviser une partition de l'espace pouvant potentiellement contenir un optimum global. A notre connaissance, il n'existe pas d'analyse en temps fini de l'algorithme DIRECT (seule la consistance a été prouvée dans Finkel and Kelley (2004)). Cependant, Munos (2014) a identifié certaines conditions de régularité locale permettant d'analyser l'algorithme SOO en temps fini. Plus précisément, en introduisant la notion de dimension effective $D \geq 0$ (définie dans l'article et dépendant à la fois du partitionnement de l'espace et de la fonction cible) qui mesure le nombre de boules disjointes permettant de couvrir l'ensemble de niveau ϵ de la fonction pour tout $\epsilon > 0$, l'auteur identifie des conditions sur les paramètres de l'algorithme permettant d'obtenir les vitesses suivantes:

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = \begin{cases} O(n^{-\frac{1}{D}}) & \text{si} \quad D > 0, \\ O(e^{-c\sqrt{n}}) & \text{si} \quad D = 0 \end{cases}$$

où $c > 0$ est une constante. Par exemple, en utilisant un partitionnement standard de l'espace de recherche et en supposant qu'il existe $x^\star \in \mathcal{X}$, $\eta, c_1, c_2, \nu > 0$ et $\alpha \geq 0$

tels que la fonction cible satisfasse $\forall x \in B(x^\star,\eta),\ \ c_1 \|x^\star - x\|^\nu \leq f(x^\star) - f(x) \leq c_2 \|x^\star - x\|^{\nu/(1+\alpha)}$ pour une semi-métrique $\|\cdot\|$ (*e.g.*, ℓ_2, ℓ_∞), ils montrent que $D = \alpha d/\nu$ lorsque $\alpha > 0$ et $D = 0$ quand $\alpha = 0$. Ainsi, l'algorithme SOO atteint dans ce cas là une vitesse de convergence polynomiale lorsque $\alpha > 0$ et une vitesse exponentielle quand $\alpha = 0$, qui est à comparer avec la vitesse d'ordre $\Omega(n^{-\nu/d})$ atteinte par la recherche aléatoire. Finalement, nous signalons que ces ces résultats ont par ailleurs été étendus au cadre d'observations bruitées dans les travaux de Valko et al. (2013) and Grill et al. (2015).

Stratégies d'évolution. (Rechenberg (1971)). Nous nous concentrons maintenant sur la classe des stratégies d'évolution-(μ,λ) qui utilisent mutation, recombinaison et sélection dans le but de séquentiellement faire évoluer l'ensemble des candidats. A notre connaissance, aucune borne supérieure sur les vitesses de convergence et aucun résultat de consistance n'ont été prouvés pour ces méthodes. Néanmoins, il est à noter que différents résultats de convergence asymptotique ont été obtenus pour cette classe d'algorithme dans les travaux de thèse de Cerf (1994) en optimisation combinatoire. Par ailleurs, Teytaud et Fournier (2008) ont réussi à obtenir des bornes inférieures exponentielles sur les vitesses de convergence pour différentes variations de l'algorithme (μ,λ)-ES en optimisation continue en utilisant la VC-dimension V des ensembles de niveaux de la fonction cible. Plus précisément, ils montrent que si V est finie, alors

$$\|X_{\hat{\imath}_n} - x^\star\|_2 = \Omega_{\mathbb{P}}(e^{-c(V)n/d})$$

où $c(V)$ est une constante qui dépend de V et de l'extension considérée. De plus, Auger (2005) a analysé la convergence de l'algorithme $(1,\lambda)$-SA-ES sur la fonction sphère $f(x) = -\|x\|_2^2$ et a identifié certaines conditions sur les paramètres de l'algorithme permettant de montrer que

$$\ln(\|X_{\hat{\imath}_n}\|_2)/n \xrightarrow{p.s.} c$$

pour une certaine constante $c \in \mathbb{R}$. Cependant, comme le signe de la limite de $\ln(\|X_{\hat{\imath}_n}\|_2)/n$ reste inconnu, ce résultat ne montre que la convergence ou divergence exponentielle de l'algorithme. Il tend toutefois à suggérer que l'algorithme peut atteindre les vitesses exponentielles obtenues dans les travaux de Teytaud et Fournier (2008) sur la fonction sphère.

Stratégies à gain espéré (EI). (Močkus (1975)). La dernière méthode que nous analysons est une stratégie bayésienne qui consiste à sélectionner à chaque itération $t \geq 2$, un point $x_{t+1} \in \arg\max_{x\in\mathcal{X}} \mathbb{E}_{f\sim\pi}[\max(f(x) - \max_{i=1\dots t} f(x_i), 0) | \{(x_i, f(x_i))\}_{i=1}^t]$ où f est supposée être distribuée selon une loi π fixée en entrée de l'algorithme. Vazquez et Bect (2010) ont montré que lorsque π est la loi d'un processus gaussien de coefficient de régularité ν (défini dans l'article) fini, la stratégie EI converge vers le maximum de n'importe quelle fonction f appartenant à l'espace de Hilbert à noyau reproduisant $\mathcal{H}_\pi$ associé à la distribution π, *i.e.*

$$\forall f \in \mathcal{H}_\pi,\ \ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x).$$

De plus, Bull (2011) a montré qu'une version adaptative de l'algorithme EI qu'ils définissent peut atteindre la vitesse polynomiale quasi-optimale suivante:

$$\forall f \in \mathcal{H}_\pi,\ \ \max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = O_{\mathbb{P}}(n^{-\nu/d})$$

lorsque π est une loi de coefficient de régularité $\nu \geq 0$. Cependant, aucune vitesse de convergence exponentielle n'a cependant été obtenue pour cette classe d'algorithmes à notre connaissance. Finalement, nous signalons que de nombreux auteurs ont analysé la convergence de ces algorithmes en présente d'évaluations bruitées dans un cadre bayésien différent où la fonction inconnue est supposée être distribuée selon une loi fixée (*e.g.*, Srinivas et al. (2010), Krause et Ong (2011) et Contal et al. (2015)).

IV Organisation du document

L'axe principal de ce document porte sur la mise en place et l'analyse de deux procédures séquentielles d'optimisation visant à identifier le maximum de n'importe quelle fonction à variations globalement régulières et irrégulières. Nous y abordons notamment les questions suivantes:

- Quelles performances peut-on espérer atteindre lorsque la fonction cible est supposée être globalement régulière et de régularité connue?
- Peut-on obtenir des performances similaires dans le cadre plus réaliste où la fonction est en effet globalement régulière mais de régularité inconnue?
- Est-il possible de mettre en place des procédures d'optimisation pour des fonctions à variations irrégulières en apprenant leurs ensembles de niveaux?

Ainsi, les travaux réalisés au cours de cette thèse ont été découpés en quatre parties. La première partie est consacrée à une présentation générale du cadre de l'étude tandis que l'ensemble des contributions originales sont regroupées dans les chapitres suivants.

Le **Chapitre 1** contient une revue de la littérature de l'optimisation globale suivie d'un résumé détaillé des contributions de la thèse. Après une brève introduction du contexte, nous y présentons les caractéristiques du problème d'optimisation que nous considérons ainsi qu'une série d'exemples applicatifs nécessitant la résolution de ce type de problème. Nous réalisons ensuite une synthèse des différents travaux et résultats théoriques documentés dans la littérature pour finalement présenter un résumé détaillé des contributions de la thèse.

Dans le **Chapitre 2**, on aborde la question de l'optimisation de fonctions régulières. L'objectif est de développer des procédures efficaces pour l'optimisation de fonctions présentant une constante de Lipschitz finie. Nous commençons par identifier une condition nécessaire et suffisante pour l'obtention de la convergence asymptotique de n'importe quelle procédure séquentielle, puis nous formulons un résultat minimax sur leurs erreurs d'approximation. Nous introduisons ensuite une première stratégie, nommée LIPO, visant à optimiser n'importe quelle fonction de constante de Lipschitz connue. Différents résultats de convergence et des bornes sur l'erreur d'approximation sont obtenus. Pour finir, nous étendons la stratégie au cas plus réaliste où la constante de Lipschitz n'est pas présupposée connue, et nous montrons que cette extension de l'algorithme présente des garanties théoriques similaires à celles de la version non adaptative.

Le **Chapitre 3** se concentre sur la mise en place de stratégies d'optimisation n'exploitant pas la régularité de la fonction inconnue. Dans un premier temps, nous observons qu'optimiser une fonction revient fondamentalement à apprendre la règle d'ordonnancement binaire qu'elle définit, ce qui nous permet de faire le lien entre théories de l'optimisation et de l'ordonnancement binaire. Nous y introduisons ensuite deux nouvelles stratégies d'optimisation reposant sur l'utilisation de n'importe quelle méthode d'ordonnancement. Différentes vitesses de convergence et résultats de consistance sont obtenus pour ces méthodes. Finalement, nous présentons une série de résultats additionnels permettant d'implémenter ces stratégies d'optimisation en pratique pour certaines structures d'ordonnancement.

Dans le **Chapitre 4**, on procède à une comparaison empirique de neuf stratégies d'optimisation globale. Les algorithmes développés dans les Chapitres 2 et 3 sont comparés avec sept autres méthodes issues de différentes approches de l'optimisation globale sur un banc d'essais comprenant des problèmes de calibration de systèmes d'apprentissage et des problèmes synthétiques fréquemment rencontrés en optimisation. Au total, 200 000 problèmes d'optimisation auront été résolus. En conclusion, nous observons que les méthodes développées dans cette thèse sont compétitives sur la plupart des problèmes, mais présentent néanmoins certaines limites. Pour clore le chapitre, nous proposons des pistes de développement visant à améliorer la stabilité et la rapidité de nos méthodes.

Enfin, l'**Annexe** contient un ensemble de résultats utilisés à de multiples reprises dans le corps du document ainsi que le résumé d'un travail préliminaire de recherche. Dans un premier temps, nous introduisons une série d'inégalités géométriques fréquemment utilisées dans les preuves des Chapitres 2 et 3. Ensuite, nous analysons et formulons un ensemble de résultats sur la convergence de la Recherche Purement Aléatoire. Finalement, nous y présentons un travail préliminaire que nous avons effectué dans un cadre de travail différent de celui du reste du document où la fonction cible est supposée être la réalisation d'une variable aléatoire de loi connue et souvent référencés d'optimisation bayésienne.

V Résumé des contributions

Comme énoncé précédemment, la contribution principale de cette thèse peut se résumer à (i) la mise en place, (ii) l'analyse théorique et (iii) la comparaison empirique de deux nouvelles stratégies séquentielles pour l'optimisation globale: AdaLIPO (Chapitre 2) et AdaRankOpt (Chapitre 3) qui sont comparées aux algorithmes de l'état de l'art dans le Chapitre 4.

V. 1 Structure générique des stratégies d'optimisation

Dans un premier temps, nous commençons par présenter la structure générique d'optimisation qui est commune à nos deux stratégies.

Optimisation sur une classe de fonctions fixée. Le premier algorithme que nous présentons (Figure V.1) vise à optimiser n'importe quelle fonction inconnue f appartenant à une classe de fonctions $\mathcal{F}$ fixée. Il prend en entrée une classe $\mathcal{F}$, un

nombre $n \geq 1$ d'itérations, l'espace d'entrée $\mathcal{X} \subset \mathbb{R}^d$ et suppose que la fonction cible $f : \mathcal{X} \to \mathbb{R}$ appartient à $\mathcal{F}$. La stratégie commence par évaluer la fonction $f(X_1)$ en un point X_1 uniformément distribué sur l'espace de recherche et définit l'ensemble actif $\mathcal{F}_1 = \{g \in \mathcal{F} : g(X_1) = f(X_1)\}$ de fonctions dans $\mathcal{F}$ qui sont consistantes avec $f(X_1)$. A chaque itération $t < n$, un point X_{t+1} est tiré uniformément sur l'espace de recherche $\mathcal{X}$ et l'algorithme décide ou non d'évaluer la fonction en ce point. Il évalue $f(X_{t+1})$ si et seulement si il existe une fonction g dans l'ensemble actif $\mathcal{F}_t := \{g \in \mathcal{F} : g(X_i) = f(X_i),\ \forall i \leq t\}$ des fonctions dans $\mathcal{F}$ qui sont consistantes avec les évaluations et qui puisse retourner une évaluation $g(X_{t+1})$ au moins supérieure ou égale à la meilleure évaluation observée jusqu'à présent $\max_{i=1\dots t} f(X_i)$. Plus précisément, le mécanisme de la règle de décision peut être expliqué comme suit: étant donné que la fonction inconnue f appartient nécessairement à l'ensemble actif $\mathcal{F}$, s'il n'existe aucune fonction $g \in \mathcal{F}_t$ telle que $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$, alors nous avons nécessairement $f(X_{t+1}) < \max_{i=1\dots t} f(X_i)$. Ainsi, l'algorithme n'évalue la fonction que sur des points qui ont une chance de retourner une évaluation au moins supérieure ou égale à la meilleure évaluation observée jusqu'à présent.

Entrée: $n \in \mathbb{N}^\star$, $\mathcal{F}$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \mathcal{F}$

1. Initialisation: Soit $X_1 \sim \mathcal{U}(\mathcal{X})$
Évaluation de $f(X_1)$, $t \leftarrow 1$
Soit $\mathcal{F}_1 = \{g \in \mathcal{F} : g(X_1) = f(X_1)\}$

2. Itérations: Tant que $t < n$:
Soit $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
S'il existe $g \in \mathcal{F}_t$ telle que $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$
Évaluation de $f(X_{t+1})$, $t \leftarrow t+1$
Soit $\mathcal{F}_t = \{g \in \mathcal{F}_{t-1} : g(X_t) = f(X_t)\}$

3. Sortie: Renvoi de $X_{\hat{\imath}_n}$ où $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure V.1: Optimisation d'une fonction inconnue f appartenant à la classe $\mathcal{F}$

La stratégie adaptative. En pratique, il est généralement impossible d'avoir une information a priori sur la fonction à optimiser et donc de savoir qu'elle appartient à un ensemble de fonctions $\mathcal{F}$ fixé. L'algorithme adaptatif (Figure V.2) est une extension directe de l'algorithme précédent qui incorpore les principes de minimisation structurelle du risque empirique de Vapnik (1992), qui est une variante de la sélection de modèle avec des classes emboîtées (suite croissante de classes de fonctions). A la place de la connaissance d'un ensemble $\mathcal{F}$, il prend en entrée un paramètre $p \in (0,1)$ et une séquence d'ensembles de fonctions emboîtés $\mathcal{F}_{k\in\mathbb{Z}}$ satisfaisant

$$\cdots \subseteq \mathcal{F}_i \subseteq \mathcal{F}_{i+1} \subseteq \cdots \subseteq \bigcup_{k\in\mathbb{Z}} \mathcal{F}_k. \tag{5.1}$$

L'idée derrière cette approche est la suivante: étant donné que l'optimisation de la fonction devrait être plus facile si la fonction inconnue est simple (*i.e.* appartient à un petit ensemble de fonctions), alors, on devrait toujours considérer à chaque

étape que la fonction inconnue appartient au plus petit ensemble de fonctions de la séquence ayant une chance de la contenir. Plus précisément, l'algorithme commence par évaluer la fonction en un point X_1 uniformément distribué sur $\mathcal{X}$ et définit l'indice $\hat{I}_1 = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ t.q. } g(X_1) = f(X_1)\}$ du plus petit ensemble de la séquence pouvant contenir f. A chaque itération $t < n$, une variable suivant une loi de Bernoulli de paramètre p est tirée. Si $B_{t+1} = 1$, l'algorithme explore l'espace en évaluant la fonction en un point uniformément distribué sur $\mathcal{X}$. Si $B_{t+1} = 0$, l'algorithme exploite les précédentes évaluations en faisant une itération du premier algorithme avec en entrée le plus petit ensemble de fonctions $\mathcal{F}_{\hat{I}_t}$ de la séquence pouvant contenir f. Lorsqu'une nouvelle évaluation $f(X_{t+1})$ a été faite, l'indice $\hat{I}_t = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ t.q. } g(X_i) = f(X_i),\ \forall i \leq t\}$ du plus petit ensemble de fonctions de la séquence $\mathcal{F}_{k\in\mathbb{Z}}$ qui peut contenir la fonction cible est mis à jour. Ainsi, le paramètre p contrôle ici le compromis entre la phase d'exploitation et la phase d'exploration qui empêche l'algorithme d'être bloqué dans un optimum local.

Entrée: $n \in \mathbb{N}^\star$, $\{\mathcal{F}_k\}_{k\in\mathbb{Z}}$, $\mathcal{X} \subset \mathbb{R}^d$, $p \in (0,1)$ $f : \mathcal{X} \to \mathbb{R}$

1. Initialisation: Soit $X_1 \sim \mathcal{U}(\mathcal{X})$
Évaluation de $f(X_1)$, $t \leftarrow 1$
Soit $\hat{I}_1 = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ t.q. } g(X_1) = f(X_1)\}$
Soit $\widehat{\mathcal{F}}_{I_1} = \{g \in \mathcal{F}_{I_1} : g(X_1) = f(X_1)\}$

2. Itérations: Tant que $t < n$:
Soit $B_{t+1} \sim \mathcal{B}(p)$
Si $B_{t+1} = 0$
Soit $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
Si $B_{t+1} = 1$
Tirage de $X_{t+1} \sim \mathcal{U}(\mathcal{X})$ jusqu'à ce que $\exists g \in \widehat{\mathcal{F}}_{I_t}$
t.q. $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$

Évaluation de $f(X_{t+1})$, $t \leftarrow t+1$
Soit $\hat{I}_t = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ t.q. } g(X_i) = f(X_i),\ \forall i \leq t\}$
Soit $\widehat{\mathcal{F}}_{I_t} = \{g \in \mathcal{F}_{I_t} : g(X_i) = f(X_i),\ \forall i \leq t\}$

3. Sortie: Renvoi de $X_{\hat{\imath}_n}$ où $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure V.2: Algorithme adaptatif sur une séquence $\{\mathcal{F}_k\}_{k\in\mathbb{Z}}$ avec un paramètre $p \in (0,1)$

Application à l'optimisation de fonctions à variations régulières. La second chapitre de cette thèse est dédié à l'application et à l'analyse de ces algorithmes pour l'optimisation de fonctions lipschitziennes. Dans ce cas-là, le premier algorithme est implémenté avec la classe de fonction $\mathcal{F}$ fixée à $\mathrm{Lip}(k)$ pour un certain $k \geq 0$ et l'algorithme adaptatif effectue la sélection de modèle sur la séquence d'ensembles de fonctions:

$$\mathrm{Lip}(0) \subseteq \cdots \subseteq \mathrm{Lip}(k_i) \subseteq \mathrm{Lip}(k_{i+1}) \subseteq \cdots \subseteq \bigcup_{k\geq 0} \mathrm{Lip}(k)$$

indexée par une suite croissante $k_{i\in\mathbb{Z}}$ de constantes de Lipschitz définissant un maillage de $\mathbb{R}^+$ (*i.e.* telle que $\forall x \in \mathbb{R}$, $\exists i \in \mathbb{Z}$ avec $k_i \leq x < k_{i+1}$). Pour ces stratégies, nous prouvons des résultats de consistance, des bornes supérieures polynomiales sur la vitesse de convergence ainsi que des vitesses rapides en introduisant une condition de régularité hölderienne. L'ensemble de ces résultats sont résumés dans la section suivante.

Application à l'optimisation de fonctions à variations irrégulières. Le troisième chapitre de cette thèse est consacré à l'analyse et l'implémentation des algorithmes d'optimisation présentés précédemment pour des fonctions à variations non-régulières. Dans ce cadre, le premier algorithme est implémenté avec l'ensemble de fonction $\mathcal{F}$ fixé à $\{f : \mathcal{X} \to \mathbb{R} \mid r_f \in \mathcal{R}\}$ où $r_f : (x,x') \mapsto \text{sgn}(f(x) - f(x'))$ correspond à la règle d'ordonnancement définie par f et $\mathcal{R}$ est une collection de règles d'ordonnancement fixée et fournie en entrée. La version adaptative de l'algorithme utilise les principes de sélection de modèle sur la séquence $\{f : \mathcal{X} \to \mathbb{R} \mid r_f \in \mathcal{R}_k\}_{k\in\mathbb{N}^*}$ indexée par une série d'ensembles emboîtés de règles d'ordonnancement:

$$\mathcal{R}_1 \subseteq \mathcal{R}_2 \subseteq \cdots \subseteq \mathcal{R}_\infty.$$

Pour ces stratégies, nous identifions des conditions permettant d'obtenir la consistance et des vitesses de convergence polynomiales sur l'erreur d'approximation de l'algorithme en introduisant une condition portant sur la régularité des ensembles de niveaux de la fonction inconnue. Un résumé de ces résultats peut être trouvé dans la suite du chapitre.

V. 2 Limitations

Avant de présenter un résumé des résultats théoriques obtenus pour ces méthodes, nous discutons ici trois limites des algorithmes d'optimisation introduits précédemment.

Implémentation. Premièrement, nous soulignons que comme ces stratégies ont été présentées de façon générique (*i.e.*, pour n'importe quel espace $\mathcal{F}$ de fonctions), certaines étapes doivent être spécifiées pour les implémenter en pratique. En particulier, les étapes de (i) maintenir l'ensemble actif $\mathcal{F}_t$ des fonctions consistantes avec les évaluations et (ii) tester s'il existe une fonction g dans cet ensemble actif pouvant renvoyer une évaluation suffisamment élevée ne sont pas détaillées et doivent être spécifiées. Cependant, il est à noter que nous proposons dans les chapitres 2 et 3, sections 2.3 et 3.5 différents moyens d'implémenter ces algorithmes pour des classes de fonctions à variations globalement régulières et irrégulières.

Évaluations bruitées. Ensuite, nous rappelons que ces stratégies ont été développées dans un cadre de données non bruitées et ne sont donc naturellement pas adaptées aux évaluations bruitées. Nous rappelons toutefois que différentes pistes sont proposées dans le document dans les chapitres 2 et 3, sections 2.3 et 3.3 pour étendre nos méthodes aux données bruitées. A titre d'exemple, étant donné un ensemble arbitraire de fonctions $\mathcal{F}$ et un échantillon $(X_1,Y_1),\ldots,(X_t,Y_t)$ de $t \geq 1$ évaluations bruitées où $Y_i = f(X_i) + \sigma\epsilon_i$ avec $\epsilon_i \sim \mathcal{N}(0,1)$, une approche générique consisterait à relaxer la définition de l'ensemble actif $\mathcal{F}_{\delta,t} = \{g \in \mathcal{F} : R_t(g) \leq \min_{g\in\mathcal{F}} R_t(g) + UB_{\delta,t}\}$ où $R_t(g) = (1/t)\sum_{i=1}^t (f(X_i) - g(X_i))^2$ correspond par exemple à l'erreur empirique

quadratique et le terme $UB_{\delta,t}$ vient de bornes standards de généralisation sur la déviation $|R_t(f) - \min_{g\in\mathcal{F}} R_t(g)|$.

Méthode de rejet en grande dimension. Finalement, il est à noter que l'utilisation de la méthode de rejet (utilisée pour la simulation de X_{t+1} dans l'algorithme présenté dans la Figure V.1) peut être limitante dans le cas d'espaces de recherche de grande dimension. En effet, étant donné l'échantillon $\{(X_i, f(X_i))\}_{i=1}^t$, on peut montrer que la complexité numérique nécessaire à la génération uniforme de $X_{t+1} \mid \{X_i\}_{i=1}^t \sim \mathcal{X}_t$ sur l'ensemble de points $\mathcal{X}_t := \{x \in \mathcal{X} : \exists g \in \mathcal{F}_t \text{ t.q. } g(x) \geq \max_{i=1\dots t} f(X_i)\}$ est bornée inférieurement, avec probabilité supérieure à $1-\delta$, par la complexité de tester s'il existe une fonction $g \in \mathcal{F}_t$ satisfaisant la règle de décision multiplié par $\lfloor \delta \cdot \mu(\mathcal{X})/\mu(\mathcal{X}_t) \rfloor$. En observant que le volume $\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\})$ de l'ensemble de niveau ϵ d'une fonction k-Lipschtzienne tend généralement à être de l'ordre de ϵ^d (*e.g.* la fonction sphère), nous en déduisons que la complexité numérique pour atteindre une précision ϵ est d'au moins $\Omega_{\mathbb{P}}(e^{d\cdot\ln(1/\epsilon)})$, et donc exponentiellement croissante avec la dimension d. Cependant, nous soulignons que différentes techniques reposant sur l'utilisation de chaîne de Markov (Reaume et al. (2001)) ou encore d'échantillonage préférentiel (Glynn and Iglehart (1989)), traditionnellement utilisées pour la simulation d'événements rares, pourraient éventuellement être utilisées pour réduire cette complexité numérique.

V. 3 Optimisation de fonctions lipschitziennes

Nous présentons ici un résumé des contributions théoriques obtenues dans le cadre de l'optimisation de fonctions à variations régulières.

Introduction. Le Chapitre 2 se focalise sur l'approche de l'optimisation globale exploitant la régularité des variations de la fonction. Cette approche de l'optimisation est justifiée par la simple observation, qu'en pratique, le système que l'on cherche à optimiser présente en effet une régularité par rapport aux entrées. Cependant, à l'exception des travaux de Munos (2014) et Grill et al. (2015) qui utilisent des mesures de régularité locales, très peu de résultats sont connus sur les propriétés des algorithmes exploitant la régularité globale de la fonction cible. L'analyse que nous fournissons dans ce chapitre considère que la fonction inconnue est de coefficient de Lipschitz fini, *i.e.*, $\exists k \geq 0$ t.q. $|f(x) - f(x')| \leq k \cdot \|x - x'\|_2$ pour tout $(x, x') \in \mathcal{X}^2$. Notre contribution est double. D'abord, nous introduisons un premier algorithme qui nécessite la connaissance de la constante de Lipschitz ainsi qu'une version adaptative qui estime la constante. Ensuite, nous fournissons des résultats de convergence en terme de consistance et des bornes sur l'erreur d'approximation en temps fini.

Résultats préliminaires. Dans le but de mettre en place des procédures efficaces, nous commençons par fournir deux résultats sur la convergence de procédures séquentielles sur la classe des fonctions lipschitziennes. Le premier résultat est une condition nécessaire et suffisante pour l'obtention de la consistance; un algorithme d'optimisation séquentiel est consistant sur l'ensemble des fonctions lipschitziennes si et seulement si:

$$\forall f \in \bigcup_{k\geq 0} \mathrm{Lip}(k), \ \sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|X_i - x\|_2 \xrightarrow{p} 0$$

où $X_1, \dots, X_n$ est une séquence de n points générés par l'algorithme sur la fonction f. Le second résultat est un résultat minimax sur la vitesse de convergence de l'erreur d'approximation sur l'ensemble des fonctions lipschitziennes avec constante de Lipschitz fixée: pour tout $n \in \mathbb{N}^\star$ et tout $k \geq 0$, nous avons

$$c_1 \cdot k \cdot n^{-\frac{1}{d}} \leq \inf_{A \in \mathcal{A}} \sup_{f \in \mathrm{Lip}(k)} \mathbb{E}\left[\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)\right] \leq c_2 \cdot k \cdot n^{-\frac{1}{d}}$$

où $\mathcal{A}$ correspond à l'ensemble des stratégies séquentielles, $c_1 > 0$ et $c_2 > 0$ sont deux constantes qui ne dépendent que de $\mathcal{X}$ et $X_1, \dots, X_n$ est une séquence de n points générés par A sur f.

Optimisation avec constante de Lipschitz fixée. L'algorithme LIPO (présenté dans la Figure 2.1, Chapitre 2) vise à optimiser n'importe quelle fonction f dans la classe $\mathrm{Lip}(k)$ pour une constante $k \geq 0$ fixée et fournie comme entrée de l'algorithme. Pour cette stratégie, nous présentons une première borne non-asymptotique sur l'erreur d'approximation: pour toute fonction $f \in \mathrm{Lip}(k)$, tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0,1)$, nous avons avec probabilité au moins supérieure à $1-\delta$,

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d}}$$

où $X_1, \dots, X_n$ est une séquence de n points générés par LIPO sur f. Cette borne prouve que l'algorithme LIPO est à la fois consistant sur le classe des fonctions k-lipschitziennes et y atteint la vitesse minimax de convergence. Par ailleurs, en introduisant une hypothèse supplémentaire, on peut obtenir des vitesses de convergence rapides: soit f une fonction k-Lipschitzienne admettant un unique optimiseur $x^\star \in \mathcal{X}$ et telle qu'il existe $\kappa \geq 1$ et $c_\kappa > 0$ pour lesquels nous ayons $f(x^\star) - f(x) \geq c_\kappa \cdot \|x^\star - x\|_2^\kappa$ pour tout $x \in \mathcal{X}$, alors, pour tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0,1)$, nous avons avec probabilité au moins égale à $1-\delta$,

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \times \begin{cases} c_1 \left(\frac{\ln(n/\delta)}{n}\right)^{\frac{\kappa}{d(\kappa-1)}} & \text{si } \kappa > 1 \\ e^{-c_2 n / \ln(n/\delta)} & \text{si } \kappa = 1. \end{cases}$$

où $c_1 > 0$ et $c_2 > 0$ sont deux constantes dépendantes de $\mathcal{X}$, k, κ et c_κ. Ces résultats sont comparés aux résultats existants dans la littérature dans le chapitre 2, sections 2.3.4 et 2.4.3.

Optimisation avec constante de Lipschitz inconnue. L'algorithme AdaLIPO (Figure 2.4 dans le Chapitre 2) est une version adaptative de l'algorithme LIPO. Il prend en entrée un paramètre $p \in (0,1)$ et une séquence de constantes de Lipschitz $k_{i \in \mathbb{Z}}$ définissant un maillage de $\mathbb{R}^+$ (*i.e.* telle que pour tout $x \in \mathbb{R}$ il existe $i \in \mathbb{Z}$ satisfaisant $k_i \leq x < k_{i+1}$). Pour cette stratégie, nous obtenons des résultats similaires à ceux de l'algorithme LIPO. Le premier résultat est une borne supérieure polynomiale sur l'erreur d'approximation de l'algorithme: pour toute $f \in \bigcup_{k \geq 0} \mathrm{Lip}(k)$, tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0,1)$, nous avons avec probabilité au moins égale à $1-\delta$:

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{5}{p} + \frac{2\ln(\delta/3)}{p\ln(1-\Gamma(f, k_{i \in \mathbb{Z}}))}\right)^{\frac{1}{d}} \cdot \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}}$$

où $\Gamma(f, k_{i\in\mathbb{Z}})$ est une constante positive, définie dans le document, qui mesure la régularité de f par rapport à $k_{i\in\mathbb{Z}}$ et $X_1, \dots, X_n$ est une séquence de n points générés par l'algorithme sur f. Comme pour LIPO, nous pouvons obtenir des vitesses de convergence rapides en utilisant la même hypothèse: pour toute fonction $f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$ admettant un unique optimiseur $x^\star \in \mathcal{X}$ et telle qu'il existe $\kappa \geq 1$ et $c_\kappa > 0$ pour lesquels nous avons $f(x^\star) - f(x) \geq c_\kappa \cdot \|x^\star - x\|_2^\kappa$ pour tout $x \in \mathcal{X}$, alors, pour tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0,1)$, nous avons avec probabilité au moins $1-\delta$:

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot C_{f,p,\delta} \times \begin{cases} c_1 \left(\frac{\ln(n/\delta)}{n}\right)^{\frac{\kappa}{d(\kappa-1)}} & \text{si } \kappa > 1 \\ e^{-c_2 n/\ln(n/\delta)} & \text{si } \kappa = 1. \end{cases}$$

où $c_1 > 0$ et $c_2 > 0$ sont deux constantes dépendantes de $\mathcal{X}$, $k_{i\in\mathbb{Z}}$, κ et c_κ et $C_{f,p,\delta}$ dépend de $f, k_{i\in\mathbb{Z}}, \delta$ et p. Ces résultats sont discutés dans le corps du document.

Perspectives et extensions. Les directions de recherche et extensions futures comprennent: (i) l'obtention de bornes inférieures dépendantes du problème considéré, (ii) l'amélioration du Théorème 2.18 énoncé dans le Chapitre 2 pour obtenir les même bornes que dans le Théorème 2.17, (iii) l'extension des algorithmes aux données bruitées, et (iv) l'identification de meilleurs choix pour les valeurs de p de la séquence de constantes de Lipschitz $k_{i\in\mathbb{Z}}$. Finalement, deux directions complètement ouvertes consisteraient à identifier (v) sous quelles conditions les algorithmes exploitant la régularité globale de la fonction sont plus performants que les algorithmes exploitant uniquement la régularité locale, et (vi) comment ces algorithmes s'adaptent à la dimension de l'espace de recherche en pratique.

V. 4 Optimisation de fonctions à variations irrégulières et structure d'ordonnancement

Nous présentons ici un résumé des principaux résultats théoriques obtenus dans le cadre de l'optimisation de fonctions à variations irrégulières.

Introduction. Dans le Chapitre 3, nous nous proposons d'explorer les concepts issus de la théorie de l'ordonnancement et de l'empilement de classifieurs Clémençon et Vayatis (2010) pour mettre en place des stratégies d'optimisation qui ne reposent pas sur la régularité de la fonction inconnue. L'idée derrière cette approche est simple: même si la fonction cible présente des variations arbitrairement grandes, la plupart de l'information nécessaire à l'identification de son maximum est contenue dans la structure d'ordonnancement qu'elle définit, i.e. comment les ensembles de niveaux de la fonction sont emboîtés les uns dans les autres. Pour exploiter cette idée, nous introduisons un nouveau paradigme d'optimisation où la complexité de la fonction est caractérisée par la règle d'ordonnancement binaire qu'elle définit. Cependant, nous soulignons que des méthodes d'optimisation voisines reposant sur l'utilisation de méthodes de classification ont été proposées dans les travaux de Robertson et al. (2013) et Yu et al. (2016) et que différents auteurs se sont attaqués à des problèmes voisins d'estimation d'ensembles de niveaux dans le cadre de la régression ou de l'estimation de densité (voir, *e.g.,* Tsybakov (1997); Cavalier (1997); Willett and Nowak (2007)). Au vu de ces travaux, notre contribution est double. Premièrement, nous introduisons deux nouveaux algorithmes

d'optimisation qui apprennent séquentiellement la règle d'ordonnancement induite par la fonction inconnue, puis, nous fournissons des résultats de convergence en terme de convergence asymptotique et en temps fini vers le maximum.

Structure d'ordonnancement. Nous commençons par introduire les concepts qui sont au coeur de notre stratégie. Nous définissons la règle d'ordonnancement induite par la fonction cible pour ensuite en déduire une mesure de complexité par l'intermédiaire de structures d'ordonnancement. La règle d'ordonnancement $r_f : \mathcal{X} \times \mathcal{X} \to \{-1,0,1\}$ induite par la fonction $f : \mathcal{X} \to \mathbb{R}$ est définie comme suit:

$$r_f(x,x') = \begin{cases} +1 & \text{if } f(x) > f(x') \\ 0 & \text{if } f(x) = f(x') \\ -1 & \text{if } f(x) < f(x') \end{cases}$$

pour tout $(x,x') \in \mathcal{X}^2$. L'argument clé de notre approche est que même si la fonction f présente des variations arbitrairement grandes, voire des discontinuités, la règle d'ordonnancement qu'elle définit peut rester simple. Nous introduisons alors la notion de structure d'ordonnancement (*i.e.* une collection de règles d'ordonnancement) dans le but de définir la complexité de la fonction cible f à travers r_f. Après avoir introduit la structure d'ordonnancement continue $\mathcal{R}_\infty := \{r_g : g \in \mathcal{C}^0(\mathcal{X})\}$, nous introduisons des structures d'ordonnancement plus restreintes comme la classe des règles d'ordonnancement polynomiales de degré $k \geq 1$ qui est définie comme suit:

$$\mathcal{R}_{\mathcal{P}_k} := \{r_h : (x,x') \mapsto \operatorname{sgn}(h(x) - h(x')) \mid h \in \mathcal{P}_k(\mathcal{X},\mathbb{R})\}.$$

Dans ce cadre-là, l'indice $k^\star = \min\{k \in \mathbb{N}^\star : r_f \in \mathcal{R}_{\mathcal{P}_k}\}$ de la plus petite structure d'ordonnancement contenant la règle r_f induite par la fonction cible définit sa complexité.

Optimisation sur une structure d'ordonnancement fixée. L'algorithme RankOpt (Figure 3.3 dans le Chapitre 3) vise à optimiser n'importe quelle fonction f qui induit une règle r_f dans une structure d'ordonnancement $\mathcal{R}$ fournie comme entrée de l'algorithme. Pour cet algorithme, nous identifions une condition minimale d'obtention de la consistance: si la fonction f induit une règle r_f dans la classe $\mathcal{R}$ et a un maximum identifiable (*i.e.* $\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}) > 0$ pour tout $\epsilon > 0$), alors nous avons

$$\max_{i=1\ldots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x)$$

où $X_1, \ldots X_n$ est une séquence de n points générés par l'algorithme sur f. En introduisant une condition qui contrôle la régularité des ensembles de niveaux de la fonction autour de son maximum, on peut obtenir des vitesses de convergence non asymptotiques: pour toute fonction f définissant une règle d'ordonnancement r_f appartenant à $\mathcal{R}$, admettant un unique optimseur $x^\star$ et ayant des ensembles de niveaux (c_α,α)-réguliers (i.e. t.q. $\exists c_\alpha > 0$ et $\alpha \geq 0$ pour lesquels $\inf_{\{x\in\mathcal{X}:f(x)=y\}} \|x^\star - x\| \leq c_\alpha \cdot \sup_{\{x\in\mathcal{X}:f(x)=y\}} \|x^\star - x\|_2^{1/(1+\alpha)}$ pour tout $y \in \operatorname{Im}(f)$), alors, pour tout $n \in \mathbb{N}$ et $\delta \in (0,1)$, nous avons avec probabilité supérieure à $1-\delta$:

$$\|x^\star - X_{\hat{\imath}_n}\|_2 \leq C_\alpha \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$

où $X_{\hat{i}_n}$ est la sortie de l'algorithme et $C_\alpha > 0$ est une constante dépendante de α et $\mathcal{X}$. En complément des ces résultats, une borne inférieure exponentielle sur l'erreur d'approximation est fournie dans le corps du document.

L'algorithme adaptatif. L'algorithme AdaRankOpt (Figure 3.4 dans le Chapitre 3) est une version adaptative de l'algorithme RankOpt qui prend en entrée un paramètre $p \in (0,1)$ et une séquence de structures d'ordonnancement emboîtées: $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \cdots \subset \mathcal{R}_\infty$. Pour cet algorithme, nous pouvons obtenir une borne supérieure sur la vitesse de convergence en utilisant la complexité de Rademacher d'une structure d'ordonnancement $\mathcal{R}$ par rapport à la règle r_f précédemment introduite dans Clémençon et al. (2010) et définie par:

$$R_n(\mathcal{R}) := \mathbb{E}\left[\sup_{r\in\mathcal{R}} \frac{1}{\lfloor n/2 \rfloor} \left| \sum_{i=1}^{\lfloor n/2 \rfloor} \epsilon_i \mathbb{I}\left\{ r(X_i, X_{i+\lfloor n/2 \rfloor}) \neq r_f(X_i, X_{i+\lfloor n/2 \rfloor}) \right\} \right| \right]$$

où $\epsilon_1, \ldots, \epsilon_{\lfloor n/2 \rfloor}$ sont $\lfloor n/2 \rfloor$ variables de Rademacher (variables signes aléatoire) et $X_1 \ldots, X_n$ sont n variables aléatoires indépendantes et uniformément distribuées sur $\mathcal{X}$. Plus précisément, en supposant que (i) l'indice $N^\star = \min\{k \in \mathbb{N}^\star : r_f \in \mathcal{R}_k\}$ de la plus petite structure d'ordonnancement de la séquence contenant r_f est fini, (ii) il existe $K > 0$ tel que $R_n(\mathcal{R}_{N^\star}) \leq \sqrt{K/n}$ pour tout $n > 0$ et (iii) la fonction a un unique optimiseur et des ensembles de niveaux (c_α, α)-réguliers, alors, pour tout $n \in \mathbb{N}^\star$ et tout $\delta \in (0,1)$, nous avons avec probabilité supérieure à $1 - \delta$:

$$\|x^\star - X_{\hat{i}_n}\| \leq C_\alpha \cdot \left(\frac{11(4 + \ln(4/\delta))}{p \inf_{r\in\mathcal{R}_{N^\star-1}} L(r)^2} \right) \cdot \left(\frac{\ln(1/\delta)}{n} \right)^{\frac{1}{d(1+\alpha)^2}}$$

où $X_{\hat{i}_n}$ correspond à la sortie de l'algorithme AdaRankOpt après n itérations et $L(r) = \mathbb{P}(r(X, X') \neq r_f(X, X'))$ correspond à l'erreur d'ordonnancement où $(X, X') \sim \mathcal{U}(\mathcal{X} \times \mathcal{X})$. En complément de ces bornes, une série de résultats permettant d'implémenter l'algorithme pour des structures d'ordonnancement spécifiques est présenté dans le document.

Perspectives du travail. Les directions de recherche et extensions futures incluent (i) l'affinement de la caractérisation d'une fonction par rapport à une structure d'ordonnancement dans le but d'identifier la classe de fonctions sur laquelle l'algorithme atteint les vitesses de convergence exponentielles, (ii) l'identification de meilleures valeurs pour le paramètre p contrôlant le compromis exploitation-exploration de l'algorithme ainsi que pour la séquence de structure d'ordonnancement utilisée pour la sélection de modèle, et (iii) l'extension des algorithmes au cadre de données bruitées. Enfin, une direction de recherche complètement ouverte serait de développer une version bayésienne de l'algorithme en fixant une loi a priori sur la règle d'ordonnancement induite par la fonction inconnue.

V. 5 Expériences numériques

Introduction. Le dernier chapitre de ce document est entièrement dédié à une comparaison empirique de neuf implémentations de stratégies d'optimisation globale. Les algorithmes AdaLIPO et AdaRank sont comparés à sept méthodes de l'état de l'art sur un

ensemble de problèmes faisant intervenir à la fois des données réelles et des fonctions synthétiques. Les algorithmes sont classés selon différents critères, comme leur capacité à identifier une solution quasi-optimale pour différents seuils de précision. Nous en déduisons que les algorithmes développés au cours de cette thèse sont compétitifs avec l'ensemble des méthodes de l'état de l'art lorsqu'il s'agit d'identifier une solution quasi-optimale mais pourraient être améliorés pour l'identification de solutions très précises de l'optimum. Enfin, différentes pistes visant à améliorer les performances empiriques de nos méthodes sont présentées à la fin du chapitre.

Protocole. Les algorithmes AdaRankOpt (Chapitre 2) et AdaLIPO (Chapitre 3) sont comparés avec sept stratégies issues de différentes approches de l'optimisation globale: BayesOpt (stratégie bayésienne, Martinez-Cantin (2014)), CMA-ES (algorithme génétique, Hansen (2006)), CRS (variation de la recherche purement aléatoire incluant des mutations locales, Kaelo and Ali (2006)), DIRECT (technique de partitionnement, Jones et al. (1993)), IHR (méthode générant une direction aléatoire à chaque étape, Zabinsky et al. (1993)), MLSL (série d'optimisations locales, Kan and Timmer (1987)) et PRS (recherche purement aléatoire, Brooks (1958)). Le banc d'essais comprend deux séries de problèmes de calibration de systèmes d'apprentissage (i.e. calibration de régressions à noyau et de réseaux de neurones) et deux séries de problèmes synthétiques fréquemment rencontrés en optimisation globale. Pour chaque fonction cible f et chaque paramètre de précision t =90%, 95% et 99%, nous avons lancé $K = 100$ fois chaque algorithme A avec un budget total de $n = 1000$ évaluations. Nous avons ensuite collecté les temps d'arrêts correspondant au nombre d'évaluations nécessaire pour atteindre une valeur cible spécifique:

$$\tau(A,f,k,t) = \min\{i = 1,\dots,n : f(X_i^{(k)}) \geq f_{\text{target}}(t)\}$$

où $f(X_i^{(k)}),\dots,f(X_n^{(k)})$ est une séquence de n points générés par l'algorithme A lors de la k-ième expérience et $f_{\text{target}}(t)$ est une valeur cible dépendant de t définie dans le document. A l'aide de ces temps d'arrêts, nous avons mesuré la performance des algorithmes à travers une collection d'indicateurs, tels que la proportion d'expériences ayant atteint la cible en fonction du nombre d'évaluations et le nombre moyen d'itérations nécessaire pour atteindre la valeur cible:

$$\forall i \leq n,\ \widehat{\mathbb{P}}(\tau < i) := \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\{\tau(A,f,k,t) < i\}, \quad \widehat{\tau}_K := \frac{1}{K}\sum_{k=1}^{K} \tau(A,f,k,t).$$

Résultats. Les performances des algorithmes ont été agrégées et comparées sur chaque série d'expériences. Nous commençons par observer qu'aucun algorithme n'est uniformément plus performant que le reste des méthodes sur l'ensemble des problèmes considérés. Cependant, l'algorithme MLSL obtient plusieurs fois les meilleurs résultats sur les problème de calage de paramètres de régressions et sur la seconde série de problèmes synthétiques en terme de vitesse de convergence. Les algorithmes DIRECT et CRS obtiennent quant à eux plusieurs fois les meilleurs résultats sur la première série de problèmes synthétiques et sur les problèmes de calibration de réseaux de neurones. Nous observons par ailleurs que les algorithmes BayesOpt et MLSL ont besoin, en moyenne, de moins d'itérations que les méthodes CMA-ES, CRS et IHR pour identifier une solution quasi-optimale. Les algorithmes développés dans cette thèse, AdaLIPO et AdaRank, sont généralement classés entre ces deux groupes d'algorithmes (voir, e.g., Figure V.3). Nous

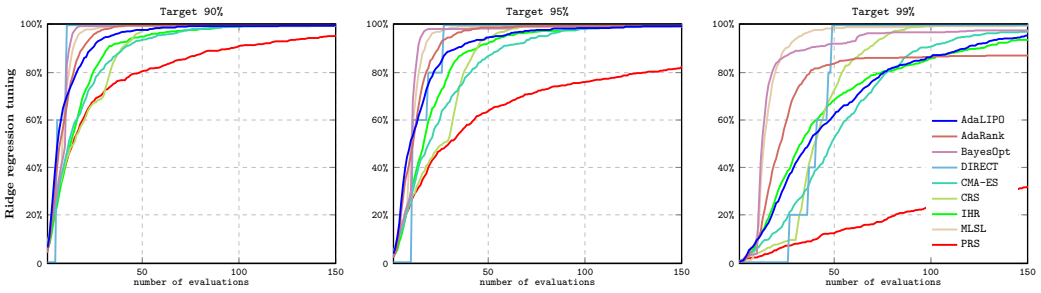


Figure V.3: Proportion du nombres d'expériences qui ont atteint les cibles 90%, 95% et 99% en fonction du nombre d'évaluations sur chacune des séries de problèmes.

nous focalisons ensuite sur le comportement de nos algorithmes et identifions certaines de leurs limites, comme le fait que AdaLIPO et AdaRank atteignent parfois la cible 95% avec très peu d'évaluations alors qu'ils ont besoin de beaucoup plus d'évaluations pour atteindre la cible 99%. Finalement, une série d'extensions visant à corriger ces limites et à améliorer leurs performances empiriques est présentée à la fin du chapitre.

Perspectives et extensions. Suite à ces expériences, nous proposons différentes extensions visant à améliorer les performances de nos méthodes. Ces extensions incluent (i) l'ajout d'un paramètre de bruit pour améliorer leur stabilité, (ii) l'incorporation de méthodes de couverture quasi-aléatoires à la place de la recherche purement aléatoire, (iii) l'incorporation de recherches locales pendant le processus d'optimisation, (iv) l'utilisation d'un compromis exploration/exploitation dynamique, fonction des évaluations, et (v) une amélioration de la procédure de la sélection de modèle. Enfin, une question complètement ouverte serait de déterminer si l'utilisation de techniques bayésiennes dans nos méthodes pourrait améliorer leurs performances sans détériorer leurs propriétés de convergence.

VI Publications associées

Finalement, l'ensemble des travaux présentés dans ce document ont donné lieu à diverses publications et communications dans des conférences internationales:

- Contal, E., Malherbe, C. and Vayatis, N. (2015). Optimization for gaussian process via chaining. *In NIPS Workshop on Bayesian Optimization.*
- Malherbe, C., Contal, E. and Vayatis, N. (2016). A ranking approach to global optimization. *In Proceedingds of the 33rd International Conference on Machine Learning (ICML), pp. 1539–1547, 2016.*
- Malherbe, C. and Vayatis, N. (2016). A ranking approach to global optimization (extended version). *ArXiv preprint:1603.04381. Submitted to JMLR.*
- Malherbe, C. and Vayatis, N. (2017). Global optimization of Lipschitz functions. *In Proceedings of the 34th International Conference on Machine Learning, PMLR 70:2314-2323, 2017.*

- Malherbe, C. and Vayatis, N. (2017). Global optimization of Lipschitz functions (extended version). *ArXiv preprint:1703.02628*.

1

Sequential global optimization

Abstract. This chapter presents a review of global optimization followed by a summary of the contributions of the thesis. After an introduction of the context, we provide a formulation of the global optimization problem that is considered throughout the document and present a series of applications that require the resolution of this type of problems. An overview of the main approaches considered in global optimization is then presented, as well as a listing of the results documented in the literature. Finally, the main contributions of the thesis are summarized at the end of the chapter.

Contents

1.1 Introduction

The occurrence of optimization problems appears to be quite frequent in industrial applications. In its simplest form, an optimization problem consists of maximizing or minimizing the output value of an objective function by sequentially choosing an input within an admissible set and computing the value of the function at this point. An important step in the optimization process is to classify the problem, since optimization algorithms are tailored to a particular type of problem, depending on the type of the objective function and input space. In particular, there is a large body of work that have been devoted to the optimization of linear, quadratic or convex functions and many software implementations aiming at solving these types of problems have also been proposed. In this work, we consider the problem of sequentially optimizing the output of an unknown and potentially nonconvex function over a continuous and bounded set. This generic problem is often referred to in the literature as global optimization (Pintér (1991)), black-box optimization (Jones et al. (1998)), non-linear optimization (Avriel (2003)) or even derivative-free optimization (Rios and Sahinidis (2013)). These problems are of particular interest when the function values are typically expensive to compute, and its derivative are not available either because the evaluations result from some physical measure, or, more commonly, because it is the result of a possibly large computer simulations. This covers in particular the problem of tuning the hyperparameters of large neural nets. Another typical example is the case of explicit objective functions that do not satisfy the standard properties used in optimization such as linearity or convexity. The major difference with the standardized problems is that the system may present more than one locally optimal point, and the classical methods of optimization which aim at finding local optima do not necessarily converge to the global optima. Nonetheless, there is still a large number of methods based on various heuristics which been introduced in order to address these problems, such as genetic algorithms (Rechenberg (1971)), Bayesian methods (Kushner (1964)) or partitioning techniques (Jones et al. (1993)). However, the major drawback of these methods is that, expect from few works, very little is known about their convergence properties. The primary purpose of this work is to address the following questions: (i) what type of nonconvex functions can we expect to optimize and (ii) which measure defines the intrinsic complexity of an optimization problem. To answer these questions, we introduce two novel sequential strategies for global optimization which aim at optimizing both smooth and non-smooth functions and provide mathematical results in terms statistical consistency and convergence toward the the optimum through the introduction of novel concepts adapted from the ranking theory (Clémençon and Vayatis (2010)) and active learning literature (Dasgupta (2011); Hanneke (2011)).

Examples. To illustrate the scope of global optimization, we now introduce a series of applications that require the resolution of a nonconvex optimization problem.

Example 1.1. (Industrial design). *This problem, taken from Sarkar et al. (2016), is the typical example of black-box optimization. It consists in finding the coordinates* $(x_1^\star, y_1^\star), \ldots, (x_m^\star, y_m^\star)$ *of* $m = 40$ *Wave Energy Converters placed in the sea which maximize the total amount of energy produced. For a single set of coordinates* $(x_1, y_1), \ldots, (x_m, y_m)$*, the amount of energy produced* $f((x_1, y_1), \ldots, (x_m, y_m))$ *is estimated numerically by solving a*

set of partial differential equations. Formally, this problem can be written as follows:

$$(x_1^\star, y_1^\star), \dots, (x_m^\star, y_m^\star) \in \underset{(x_i,y_i)_{i=1}^m \in [0,1]^{2m}}{\arg\max} f((x_1,y_1),\dots,(x_m,y_m))$$

where evaluating the function requires heavy numerical simulations with significant computational cost. Hence, the exploration of the search space can only be done with small samples.

Example 1.2. (Non-linear equations). *We now consider a toy problem involving a set of non-linear equations. Suppose that one wishes to find a pair of point* $(x^\star, y^\star) \in [-2,2]^2$ *solution of the following pair of equations:*

$$\begin{aligned} x - \sin(2x+3y) - \cos(3x-5y) &= 0 \\ y - \sin(x-2y) + \cos(x+3y) &= 0 \end{aligned}$$

An approach to approximate a solution consists in minimizing the ℓ_2*-norm loss associated with the system, i.e.,*

$$(x^\star, y^\star) \in \underset{(x,y)\in[-2,2]^2}{\arg\min} (x - \sin(2x+3y) - \cos(3x-5y))^2 + (y - \sin(x-2y) + \cos(x+3y))^2.$$

In this case, evaluating the function is almost straightforward but we point out that the resolution of such problems requires the use of an optimization algorithm when there are many parameters to be to optimized.

Example 1.3. (Regression). *Suppose that we have collected a sample* $(x_1, g(x_1))$, …, $(x_n, g(x_n))$ *of* n *evaluations of a function* $g : \mathbb{R}^d \to \mathbb{R}$ *and we wish to build an estimate* $\hat{g}$ *of this function using gaussian kernel ridge regression. The computation of this estimate requires a bandwidth parameter* $\sigma > 0$ *and a smoothness parameter* $\lambda \geq 0$ *to be set as input. A standard approach consists in choosing the pair of parameters* $(\sigma^\star, \lambda^\star)$ *which minimize the generalization error of the model computed over a K-fold cross-validation, i.e.,*

$$(\sigma^\star, \lambda^\star) \in \underset{\sigma>0,\, \lambda\geq 0}{\arg\min} \sum_{k=1}^{K} \frac{1}{|I_k|} \sum_{i\in I_k} \left(g(x_i) - \hat{g}^k_{\sigma,\lambda}(x_i) \right)^2$$

where

$$\hat{g}^k_{\sigma,\lambda} \in \underset{\hat{g}\in\mathcal{H}_\sigma}{\arg\min} \frac{1}{n - |I_k|} \sum_{i\notin I_k} (g(x_i) - \hat{g}(x_i))^2 + \lambda \|\hat{g}\|_{\mathcal{H}_\sigma},$$

$I_1, \dots, I_K$ *are* K *disjoints subset of integers,* $\mathcal{H}_\sigma$ *is the gaussian RKHS of parameter* σ *and* $\|\cdot\|_{\mathcal{H}_\sigma}$ *is the corresponding norm. In practice, a solution to this problem is generally approximated by performing a grid search over logarithm of the parameters on a restricted domain, e.g.,* $(\ln(\sigma), \ln(\lambda)) \in [-5,5] \times [-5,5]$*, where the computational cost is an increasing function of the size of the dataset.*

Example 1.4. (Neural nets hyperparameter tuning). *We now consider the toy problem of estimating two hyperparameters of a neural net. More precisely, a generic task consists in estimating the learning rate* $\eta^\star$ *and the momentum* $\mu^\star$ *of an accelerated stochastic gradient descent which minimize the empirical classification error of the predictions of a neural net with fixed architecture tuned with a stochastic gradient descent over a K-fold cross validation.*

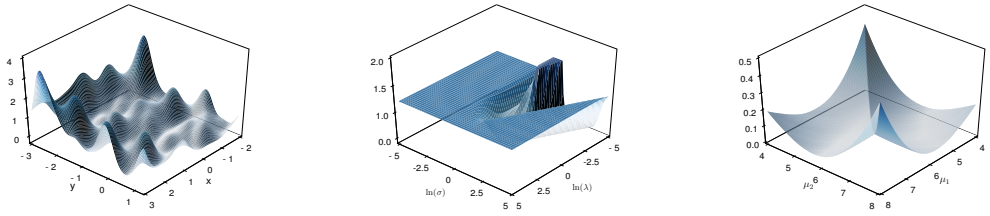


Figure 1.1: *Left:* Objective function of Example 1.2. *Middle:* Test function of Example 1.3 computed with the Abalone dataset taken from the UCI machine learning repository. *Right:* Objective function of Example 1.5 computed with the Iris dataset taken from the UCI machine learning repository.

For a data set $(x_1, y_1), \ldots, (x_n, y_n)$ *with* $x_i \in \mathbb{R}^d$ *and* $y_i \in \{0,1\}$*, the task consists in solving the following optimization problem:*

$$(\ln(\eta^\star), \ln(1-\mu^\star)) \in \underset{(\ln(\eta),\ln(1-\mu))\in\mathcal{X}}{\arg\min} \frac{1}{5}\sum_{k=1}^{5} \frac{1}{|I_k|} \sum_{i\in I_k} \mathbb{I}\Big\{\hat{g}^k_{\eta,\mu}(x_i) \neq y_i\Big\}$$

where

$$\hat{g}^k_{\eta,\mu}(x) = \mathbb{I}\Big\{\sigma_3(W_3^k \sigma_2(W_2^k \sigma_1(W_1^k x))) \geq 1/2\Big\}$$

over $\mathcal{X} = [-5, 1/2] \times [-5, 0]$ *and where* $I_1, \ldots, I_5$ *are 5 disjoint subsets of integers of similar sizes forming a partition of* $\{1, \ldots, n\}$*, the matrix* $W_1^k \in \mathbb{R}^{d\times 12}$*,* $W_2^k \in \mathbb{R}^{12\times 8}$ *and* $W_3^k \in \mathbb{R}^{8\times 1}$ *are learned with an accelerated stochastic gradient descent minimizing the cross-entropy binary loss tuned with the parameters* (η, μ) *over the data set* $\{(x_i, y_i)\}_{i\notin I_k}$ *and where* σ_1, σ_2 *and* σ_3 *are respectively the ReLu* $(\max(x, 0))$*, ReLu and sigmoid* $((1 + e^{-x})^{-1})$ *activation functions. Again, the computational time is an increasing function of the size of the network and dataset.*

Example 1.5. (Clustering I). *Suppose that we have collected a sample* $x_1, \ldots, x_n$ *of* n *points in* $\mathbb{R}^d$ *and we wish to build a decision rule* $\hat{g}_n : \mathbb{R}^d \mapsto \{1, \ldots, K\}$ *for some fixed* $K \geq 2$ *which associates to each point* x *in* $\mathbb{R}^d$ *a cluster* $\hat{g}_n(x)$*. The standard k-means method consists in using the decision rule* $\hat{g}_n : x \mapsto \arg\min_{k=1\ldots K} \left\|x - \mu_k^\star\right\|_2$ *where the centers of the clusters* $\mu_1^\star, \ldots, \mu_n^\star$ *minimize the within-cluster sum of squares, i.e.,*

$$(\mu_1^\star, \ldots, \mu_K^\star) \in \underset{(\mu_1\ldots\mu_K)\in\mathbb{R}^{d\times K}}{\arg\min} \sum_{i=1}^{n} \min_{k=1\ldots K} \|x_i - \mu_k\|_2^2.$$

In practice, this problem is generally solved with the k-means algorithm. Recall however that this algorithm does not necessarily converge towards a global optimum.

Example 1.6. (Clustering II). *Consider the same framework as in Example 1.5. A second approach consists in using a gaussian mixture to build a decision rule of the form* $\hat{g}_n(x) = \arg\min_{k=1\ldots K} \pi_k f(x \mid \mu_k, \sigma_K)$ *for some parameters* $(\pi_1, \mu_1, \sigma_1), \ldots, (\pi_K, \mu_K, \sigma_K)$ *that have to be determined and* $f(\cdot \mid \mu_k, \sigma_k)$ *denotes the gaussian cumulative distribution function. In*

this case, a natural approach consists in using the parameters that maximize the log-likelihood of the model, i.e.,

$$(\pi_1^\star,\mu_1^\star,\sigma_1^\star),\dots,(\pi_K^\star,\mu_K^\star,\sigma_K^\star) \in \underset{(\pi_i,\mu_i,\sigma_i)_{i=1}^K \in \mathcal{X}}{\arg\max} \sum_{i=1}^n \log\left(\sum_{k=1}^K \pi_k f(x_i|\mu_k,\sigma_k)\right).$$

where $\mathcal{X} = \{\pi_i \in [0,1],\ \sigma_i > 0,\ \mu_i \in \mathbb{R} :\ \sum \pi_i = 1\}$. *In practice, an EM algorithm is generally used to solve this problem, but typically converges towards a local optimum.*

Framework. As shown above, each global optimization problem presents its own distinct features. For instance, the objective function can be either totally unknown or explicit; the search space can be infinite, constrained or bounded and evaluating the function can be almost straightforward or computationally intensive. In this work, we focus on the following framework where:

- The objective function does not have to be specified and can be unknown,
- The input space is a compact and convex subset of $\mathbb{R}^d$,
- The function evaluations are assumed to be noiseless,
- No specific assumptions are made on the computational cost, we thus consider the worst case where the evaluations are expensive to compute.

However, we point out that several extensions are proposed throughout the document in order to adapt the algorithms to settings with noisy evaluations and we stress that they could be extended to unbounded domains using mapping functions (*e.g.,* $\min_{x\in\mathbb{R}} f(x) = \min_{x\in[-\pi/2,\pi/2]} f(\tan(x))$). The next section presents the detailed setup that is considered throughout the thesis as well as the associated performance criteria.

1.2 Problem statement

Setup. Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact and convex set with non-empty interior and let $f : \mathcal{X} \to \mathbb{R}$ be an unknown function which is only supposed to admit a maximum over its input domain. The goal in global optimization consists in finding some point

$$x^\star \in \underset{x\in\mathcal{X}}{\arg\max}\, f(x)$$

with a minimal amount of function evaluations. The standard setup involves a sequential procedure which starts by evaluating the function $f(X_1)$ at an initial random point $X_1 \in \mathcal{X}$ and iteratively selects at each step $t \geq 1$ an evaluation point $X_{t+1} \in \mathcal{X}$ which depends on the previous evaluations $(X_1,f(X_1)),\dots,(X_t,f(X_t))$ and is $\mathcal{F}_t$-measurable where $\mathcal{F}_t := \sigma((X_1,f(X_1)),\dots,(X_t,f(X_t)))$ denotes the filtration generated by the history of available information, and receives the evaluation of the function $f(X_{t+1})$ at this point. After n iterations, we consider that the algorithm returns (one of) the evaluation point which has recorded the highest evaluation:

$$X_{\hat{\imath}_n} \quad \text{where} \quad \hat{\imath}_n \in \underset{i=1\dots n}{\arg\max}\, f(X_i).$$

The analysis provided in the document considers that the number n of evaluation points is not fixed and it is assumed that function evaluations are noiseless. As an example, the generic structure of a sequential optimization procedure is displayed in Figure 1.4.

Input: $n \in \mathbb{N}^\star$, $\mathcal{X} \subset \mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Choose $X_1 \in \mathcal{X}$
Evaluate $f(X_1)$, $t \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Choose $X_{t+1} \in \mathcal{X}$ based on the previous evaluations $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$
Evaluate $f(X_{t+1})$, $t \leftarrow t + 1$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Figure 1.2: Generic structure of a sequential optimization algorithm

Performance metrics. The performance of an algorithm over a function f is measured after n iterations through the difference between the value of the true maximum and the value of the highest evaluation observed so far:

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i)$$

where $X_1, \ldots, X_n$ denotes a sequence of n evaluation points generated by the algorithm over f after n iterations. However, as we will see in Chapter 3, this criterion can not always be bounded when the target function is not assumed to be at least locally smooth. In that case, considering that the global optimizer $x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$ is unique, the performance of an algorithm will be measured through the Euclidean distance between the true optimizer and its approximation:

$$\|X_{\hat{\imath}_n} - x^\star\|_2 .$$

Remark that in the case where f admits a unique optimizer and is assumed to be k-Lipschitz for a some $k \geq 0$, a bound on the first performance criterion can be recovered from the second one using the following inequality: $\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq k \cdot \|X_{\hat{\imath}_n} - x^\star\|_2$. Given these performance metrics, we may now describe the corresponding concept of asymptotic convergence in the sense of global optimization.

Definition 1.7. (Optimization consistency) *A global optimization algorithm is said to be consistent over a set $\mathcal{F}$ of real-valued functions admitting a maximum over their input domain $\mathcal{X}$ if and only if*

$$\forall f \in \mathcal{F}, \quad \max_{i=1\ldots n} f(X_i) \xrightarrow{p} \max_{x \in \mathcal{X}} f(x)$$

where $X_1, \ldots, X_n$ denotes a sequence of n evaluation points generated by the algorithm over the function f.

The next definition introduces the notion of convergence rate that will be used to analyze the nonasymptotic convergence of an algorithm over a set of real-valued functions.

Definition 1.8. (Convergence rate) *A global optimization algorithm is said to converge at rate $U_{n,\delta} \to_{n\to\infty} 0$ over a set $\mathcal{F}$ of real-valued functions admitting a maximum over their input domain $\mathcal{X}$ if and only if for all $f \in \mathcal{F}$, all $\delta \in (0,1)$ and all $n \in \mathbb{N}^\star$, we have with probability at least $1-\delta$:*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq U_{n,\delta}$$

where $X_1, \dots, X_n$ denotes a sequence of n evaluation points generated by the algorithm over the function f.

The decreasing rate of the series $U_{n,\delta}$ describes the speed of convergence of the algorithm. In particular, we will use the notation $\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = O_{\mathbb{P}}(n^{-\alpha})$ and talk about polynomially decreasing rate when for any $\delta \in (0,1)$, $U_{n,\delta} = O(n^{-\alpha})$ for some $\alpha > 0$ and exponentially decreasing rate when $U_{n,\delta} = O(e^{-cn})$ for some $c > 0$.

1.3 Existing methods

Here, we present a list of the main approaches that have been considered in global optimization as well as a listing of the main theoretical results documented in the literature.

Taxonomy of optimization methods. We start by providing an overview of the main techniques considered in global optimization. We point out that the listing provided below only contains the works that are known by the author and is thus not intended to be exhaustive, and that some classes have been adapted and modified from the taxonomies of Pardalos et al. (2000) and Rios and Sahinidis (2013).

Local search methods. As the name suggests, these algorithms aim at finding a local optimizer of the function based on a starting point set as input. Even in this subclass of algorithms, many strategies have been proposed, such as the Nelder-Mead simplex algorithm (Nelder and Mead (1965)), the Generalized Pattern Search (Torczon (1997); Audet and Dennis Jr (2006)) or the quadratic searches of Powell (1994) and Conn et al. (1997). Since a global optimization problem have in general many local optima, the convergence of these methods toward the global optimum highly depends upon the starting point. On the other hand, they can often quickly locate a local optima, even in very high-dimensional problems. Local searches are thus often used after the run of a global search method to "polish" the optimum to a greater accuracy.

Covering methods. These simple strategies consist in evaluating the function over a sequence of points which ends covering the whole input space regardless of the function values. The Grid Search (Walsh (1970)) and the Pure Random Search (Brooks (1958)) are the canonical examples of covering methods. However, we point out that more sophisticated techniques relying on low-discrepancy sequences (Sobol (1967); Sobol and Levitan (1976); Niederreiter (1988)) or Latin Hypercube Sampling (McKay et al. (1979); Helton and Davis (2003)) have also been proposed. Although these methods are known

to be consistent over a large class of functions (see Appendix Section), they are generally slow in practice and more specifically when the dimensionality of the problem is large. They are often used at the beginning of an optimization process to learn the global structure of the target function.

Partitioning techniques. These methods partition the search space into subdivisions of similar sizes in order to sequentially evaluate the function and divide a subdivision of the space that might contain the optimum. Several strategies have been proposed to assign the potential of subdivision to contain the optimum. For instance, Jones et al. (1993) proposed to use the set of admissible Lipschitz constants to compute an upper bound on the function values while Munos (2014) considered local smoothness measures and Huyer and Neumaier (1999) proposed to assign to each subdivision a level which is an increasing function of the number of times a subdivision has been processed.

Model-based searches. Model-based strategies use a surrogate model of the objective function that is sequentially updated in order to guide the optimization process. The interpolating functions include the standard polynomial, sinusoidal and radial basis functions (see, *e.g.*, Barton (1994) for a review) but also stochastic models such as Kriging (Matheron (1963); Sacks et al. (1989)) or Gaussian processes (Kushner (1964); Močkus (1975); Jones et al. (1998)). In general, the approximation of the function is estimated by minimizing the sum of square deviations computed on the previous evaluations. Last, we point out that in the case of stochastic models, this subfield of global optimization is often referred to as Bayesian optimization.

Evolutionary strategies. Introduced by Holland (1975) and extended to continuous domains by Bethke (1978), evolutionary strategies use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection to generate the next evaluation points. In this setting, the evaluation points play the role of individuals in a population, and the function values determine the quality of the individuals. Recent developments in this class of algorithms include techniques that generate the next evaluations points according to a normal distribution with mean vector and covariance matrix estimated from the previous evaluations, allowing the evaluation points to adapt to the contours of the objective function (see, *e.g.*, CMA-ES, Hansen (2006)).

Hit-and-run algorithms. Proposed by Boneh and Golan (1979) and Smith (1984), each iteration of hit-and-run algorithms compares the current best evaluation point with a randomly generated candidate. The generation of the new candidate is based on two random components: a direction uniformly sampled over the unit sphere and a step size. The algorithm moves towards that direction if only if it improves the function values. We point out that several extensions of this method have been proposed, allowing arbitrary distributions for both the generation of the direction and step size (see, *e.g.*, Bélisle et al. (1993) or Zabinsky (2003)).

Simulated annealing. Simulated annealing was first proposed to handle combinatorial problems in Kirkpatrick et al. (1983). At each iteration, the algorithm generates a new evaluation point in the neighborhood of the current iterate point, and probabilistically decides between moving toward the new point depending on the function values. As

a result, unlike hit-and-run algorithms, simulated annealing allows moves towards points with objective function values worse than the current iterate. The probability of acceptance depends on a sequence of temperature parameters that is generally referred to as the cooling schedule. Last, we point out that these methods are also used to solve similar problems such as the estimation of different local optima (see, *e.g.*, Roth et al. (2015); Carr et al. (2016)).

Note that, in addition to these descriptions, a timeline in the history of innovations in global optimization can be found in Figure 1.3. We may now present a listing of the theoretical results reported in the literature.

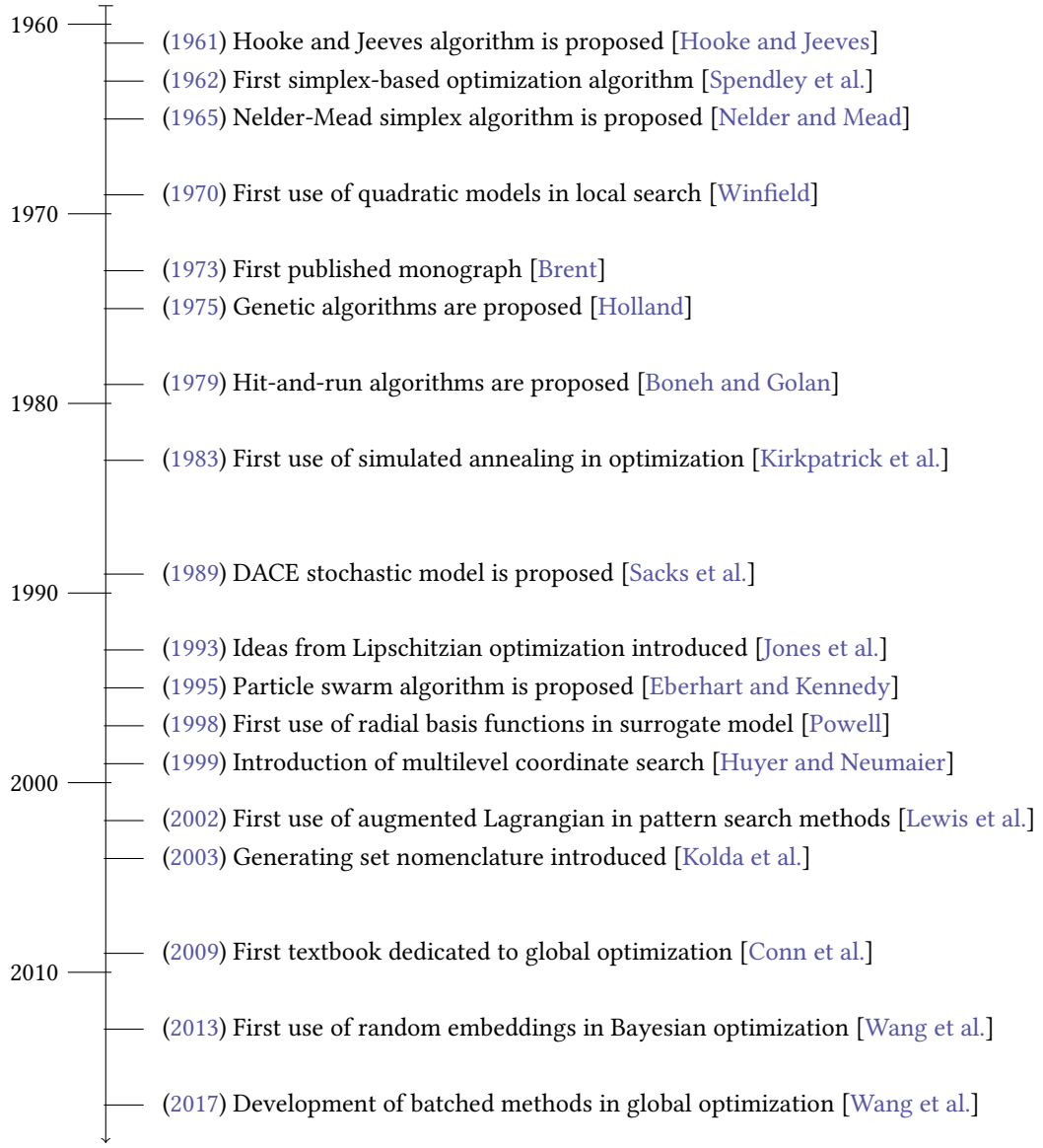


Figure 1.3: Timeline of innovation in global optimization (adapted from Rios and Sahinidis (2013)).

Convergence results. Table 1.3 collects a list of the main theoretical results documented in the literature for the different classes of algorithms presented above available in similar settings. To provide an insight on the type of convergence results that are expected in global optimization, we now present a summary of the results that are closely related to ours.

Pure Random Search. First, we consider the standard Pure Random Search which consists in evaluating the function over a sequence of points $X_1, \dots, X_n$ uniformly and independently distributed over the input space $\mathcal{X}$. Due to the simplicity of the method, one can easily analyse its convergence properties by using the ϵ-level set of the target function $\mathcal{X}_\epsilon := \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$. In particular, denoting by $\mathcal{F}_{\text{id}} = \{f : \mathcal{X} \to \mathbb{R} \text{ s.t. } \forall \epsilon > 0, \ \ \mu(\mathcal{X}_\epsilon) > 0\}$ the set of functions that have an identifiable maximum, one can easily derive the consistency property of the strategy:

$$\forall f \in \mathcal{F}_{\text{id}}, \ \ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x)$$

Moreover, assuming that the unknown function f admits a unique optimizer $x^\star \in \arg\max_{x\in\mathcal{X}} f(x)$ and that there exists $c_1, c_2 \geq 0$ and $\kappa_1 \geq \kappa_2 \geq 0$ for which we have $c_1 \cdot \|x - x^\star\|_2^{\kappa_1} \leq f(x^\star) - f(x) \leq c_2 \cdot \|x - x^\star\|_2^{\kappa_2}$, for all $x \in \mathcal{X}$, it can easily be shown that for all $\delta \in (0,1)$, we have with probability at least $1 - \delta$:

$$C_1 \cdot \left(\frac{\delta}{2n}\right)^{\frac{\kappa_1}{d}} \leq \max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq C_2 \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{\kappa_2}{d}}$$

Algorithm	Consistency	Upper bound	Lower bound
Covering			
PRS	Anderssen et al. (1975)	Anderssen et al. (1975)	Appendix Sec.
Partitioning			
DIRECT	Finkel et al. (2004)	-	-
MCS	Huyer et al. (1999)	-	-
SOO	Munos (2014)	Munos (2014)	-
Model-based			
EGO	Vazquez et al. (2010)	Bull (2011)	-
RBF	Gutmann (2001)	-	-
Evolutionary			
(μ,λ)-**ES**	-	-	Teytaud et al. (2008)
$(1,\lambda)$-**SA-ES**	Auger (2005)	-	-
Hit-and-run			
H&R	Bélisle et al. (1993)	-	-
IHR	Zabinsky (2003)	Zabinsky (2003)	-
This thesis			
AdaLIPO	Malherbe et al. (2017)	Malherbe et al. (2017)	-
AdaRank	Malherbe et al. (2016)	Malherbe et al. (2016)	-

Table 1.1: Theoretical results documented in the global optimization literature. Consistency refers to asymptotic results while upper and lower bounds refer to finite-time results. Dash symbols are used when no result could be found.

where C_1 and C_2 are two constants. Remark that these convergence rates are only polynomially decreasing with $d \geq 1$ and thus subject to the curse of dimensionality.

DIRECT and SOO (Jones et al. (1993) and Munos (2014)). These partitioning methods use a splitting technique of the search space in order to sequentially evaluate the function on subdivisions of the space that have recorded the highest evaluation among all the subdivisions of similar size. To the best of our knowledge, there is no finite-time analysis of the DIRECT algorithm (only the consistency was proven by Finkel and Kelley (2004)). However, Munos (2014) identified some local smoothness conditions allowing to derive a finite-time analysis of the SOO algorithm. Precisely, introducing the notion of near-optimality dimension $D \geq 0$, defined therein, which measures the number of disjoint balls required to to cover the ϵ-level set $\mathcal{X}_\epsilon$ of the function for all $\epsilon > 0$, they show that under specific conditions on the parameters of the algorithm:

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = \begin{cases} O(n^{-\frac{1}{D}}) & \text{if} \quad D > 0, \\ O(e^{-c\sqrt{n}}) & \text{if} \quad D = 0 \end{cases}$$

for some constant $c > 0$. As an example, assuming there exists $x^\star \in \mathcal{X}$, $\eta, c_1, c_2, \nu > 0$ and $\alpha \geq 0$ such that $\forall x \in B(x^\star, \eta),\ c_1 \|x^\star - x\|^\nu \leq f(x^\star) - f(x) \leq c_2 \|x^\star - x\|^{\nu/(1+\alpha)}$ for some semi-metric $\|\cdot\|$ (*e.g.*, ℓ_2, ℓ_∞), they show that $D = \alpha d/\nu$ when $\alpha > 0$ and $D = 0$ when $\alpha = 0$. Hence, in this case, the SOO algorithm can achieve a polynomially decreasing rate when $\alpha > 0$ and an exponential decay when $\alpha = 0$ to be compared with the rate of order $\Omega(n^{-\nu/d})$ of the Pure Random Search. Last, it is also noteworthy these results have been extended to the noisy scenario in the works of Valko et al. (2013) and Grill et al. (2015).

Evolution Strategies (Rechenberg (1971)). We now consider the class of (μ, λ)-Evolution Strategies which use mutation, recombination, and selection in order to iteratively evolve the set of evaluation points. As far as we know, no consistency results or generic upper bounds have been proven for these algorithms. However, Teytaud and Fournier (2008) were able to derive exponentially decreasing lower bounds for several extensions of the (μ, λ)-ES using the VC-dimension V of the level sets of the unknown function. Precisely, they showed that if V is finite, then

$$\|X_{\hat{\imath}_n} - x^\star\|_2 = \Omega_{\mathbb{P}}(e^{-c(V)n/d})$$

where $c(V)$ is a constant that depends on both the extension under consideration and V. Moreover, Auger (2005) also analyzed the convergence of the $(1, \lambda)$-SA-ES algorithm on the sphere function $f(x) = -\|x\|_2^2$ and proved specific conditions on the parameters of the algorithm in order to ensure that

$$\ln(\|X_{\hat{\imath}_n}\|_2)/n \xrightarrow{a.s.} c$$

for some constant $c \in \mathbb{R}$. However, since the sign of the limit of $\ln(\|X_{\hat{\imath}_n}\|_2)/n$ remains unknown, this result only proves the exponential convergence or divergence of the algorithm but tends to suggest that the algorithm can achieve, in some cases, an exponentially decreasing rate similar to the one obtained by Teytaud and Fournier (2008).

Expected Improvement Strategy (Močkus (1975)). The last algorithm we consider is a Bayesian optimization strategy which selects at each step $t \geq 2$ an evaluation point

$x_{t+1} \in \arg\max_{x\in\mathcal{X}} \mathbb{E}_{f\sim\pi}[\max(f(x) - \max_{i=1\dots t} f(x_i), 0) | \{(x_i, f(x_i))\}_{i=1}^t]$ where f is assumed to be drawn from a law π set as input. Vazquez and Bect (2010) showed that when π is a fixed Gaussian process prior with a finite smoothness, the EI strategy converges on the maximum of any function f of the reproducing kernel Hilbert space $\mathcal{H}_\pi$ canonically attached to π, *i.e.*

$$\forall f \in \mathcal{H}_\pi, \quad \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x).$$

Moreover, Bull (2011) went on to prove that an adaptive version of the EI algorithm they define could achieve a near-optimal polynomial bound:

$$\forall f \in \mathcal{H}_\pi, \quad \max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) = O_{\mathbb{P}}(n^{-\nu/d})$$

when π is a prior of smoothness parameter $\nu \geq 0$. However, we are not aware at this point of any exponentially decreasing rates for this class of algorithms. Last, we point out that several authors have also proposed a theoretical analysis of these algorithms in a different setting where the unknown is assumed to be drawn from a Gaussian process and the function evaluations are assumed to be noisy (see, *e.g.*, Srinivas et al. (2010), Krause and Ong (2011) and Contal et al. (2015)).

1.4 Contributions

The main contribution of this work is the introduction and analysis of two sequential strategies for global optimization: AdaLIPO (Chapter 2) and AdaRankOpt (Chapter 3).

1.4.1 Generic optimization scheme

We start by presenting the generic optimization structure that is behind both these algorithms.

Optimization over a class of functions. The first algorithm we present (displayed in Figure 1.4) aims at maximizing any function f in a fixed class of real-valued functions $\mathcal{F}$. It takes as input the class $\mathcal{F}$, a number n of function evaluations, the input space $\mathcal{X}$ and considers that the target function f belongs to $\mathcal{F}$. It starts by evaluating the function $f(X_1)$ over a point X_1 uniformly distributed over the input space and defines the active set $\mathcal{F}_1 = \{g \in \mathcal{F} : g(X_1) = f(X_1)\}$ of functions in $\mathcal{F}$ which are consistent with $f(X_1)$. At each iteration $t < n$, a point X_{t+1} is sampled uniformly over $\mathcal{X}$ and the algorithm decides whether or not to evaluate the function at this point. It evaluates $f(X_{t+1})$ if and only if there exists a function g in the active subset $\mathcal{F}_t := \{g \in \mathcal{F} : g(X_i) = f(X_i),\ \forall i \leq t\}$ of functions in $\mathcal{F}$ that are consistent with the sample which might return an evaluations $g(X_{t+1})$ at least equal to the best value observed so far $\max_{i=1\dots t} f(X_i)$. More precisely, the mechanism behind the decision rule can be explained as follows: since the objective function f always belongs to the active set $\mathcal{F}_t$, if there does not exists any $g \in \mathcal{F}_t$ such that $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$, it necessarily means that $f(X_{t+1}) < \max_{i=1\dots t} f(X_i)$. Hence, the algorithm only evaluates the function over points which have a chance to return an evaluation at least equal to the best evaluation observed so far.

Input: $n \in \mathbb{N}^\star$, $\mathcal{F}$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \mathcal{F}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
Let $\mathcal{F}_1 = \{g \in \mathcal{F} : g(X_1) = f(X_1)\}$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If there exists $g \in \mathcal{F}_t$ such that $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
Let $\mathcal{F}_t = \{g \in \mathcal{F}_{t-1} : g(X_t) = f(X_t)\}$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure 1.4: Optimization of an unknown function f in the class $\mathcal{F}$

Adaptive algorithm. In practice, it can generally not be assumed that the objective function f belongs to a given set $\mathcal{F}$. The adaptive algorithm (displayed in Figure 1.5) is an extension of the previous algorithm which follows the principle of Structural Risk Minimization of Vapnik (1992). Instead of a fixed set of functions $\mathcal{F}$, it takes as input a parameter $p \in (0,1)$ and a nested sequence of sets $\mathcal{F}_{k\in\mathbb{Z}}$ satisfying

$$\cdots \subseteq \mathcal{F}_i \subseteq \mathcal{F}_{i+1} \subseteq \cdots \subseteq \bigcup_{k\in\mathbb{Z}} \mathcal{F}_k. \tag{1.1}$$

The idea behind this algorithm is simple: since the optimization should be easier if the target function is simple (*i.e.* belongs to a small set of functions), then, one should always consider the smallest set of function of the sequence that probably contains f. The algorithm starts by evaluating the function at a point X_1 uniformly distributed over $\mathcal{X}$ and defines the index $\hat{I}_1 = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ s.t. } g(X_1) = f(X_1)\}$ of the smallest set of functions of the sequence which might contain f. At each iteration $t < n$, a Bernoulli random variable B_{t+1} of parameter p is sampled. If $B_{t+1} = 1$, the algorithm explores the space by evaluating the function at a point uniformly sampled over $\mathcal{X}$. If $B_{t+1} = 0$, the algorithm exploits the previous evaluations by making an iteration of the first algorithm with the smallest set of function $\mathcal{F}_{\hat{I}_t}$ of the sequence that probably contains the target function f. Once a new evaluation $f(X_{t+1})$ has been made, the index $\hat{I}_t = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ s.t. } g(X_i) = f(X_i),\ \forall i \leq t\}$ of the smallest set of functions of the sequence $\mathcal{F}_{k\in\mathbb{Z}}$ containing a function consistent with the sample is updated. Thus, the parameter p drives the trade-off between the exploitation and the exploration phase which prevents the algorithm from getting stuck in a local maximum.

Application to smooth functions. The second chapter of this thesis is dedicated to the implementation and analysis of the optimization scheme applied to the class of Lipschitz functions. In this setting, the first algorithm is implemented with a set $\mathcal{F}$ set to $\mathrm{Lip}(k)$ for a given $k \geq 0$ and the adaptive algorithm performs model selection over the nested sequence:

$$\mathrm{Lip}(0) \subseteq \cdots \subseteq \mathrm{Lip}(k_i) \subseteq \mathrm{Lip}(k_{i+1}) \subseteq \cdots \subseteq \bigcup_{k\geq 0} \mathrm{Lip}(k)$$

Input: $n \in \mathbb{N}^\star$, $\{\mathcal{F}_k\}_{k\in\mathbb{Z}}$, $\mathcal{X} \subset \mathbb{R}^d$, $p \in (0,1)$ $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
Let $\hat{I}_1 = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ s.t. } g(X_1) = f(X_1)\}$
Let $\widehat{\mathcal{F}}_{I_1} = \{g \in \mathcal{F}_{I_1} : g(X_1) = f(X_1)\}$

2. Iterations: Repeat while $t < n$:
Let $B_{t+1} \sim \mathcal{B}(p)$
If $B_{t+1} = 0$
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $B_{t+1} = 1$
Sample $X_{t+1} \sim \mathcal{U}(\mathcal{X})$ until $\exists g \in \widehat{\mathcal{F}}_{I_t}$ s.t. $g(X_{t+1}) \geq \max_{i=1\dots t} f(X_i)$

Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
Let $\hat{I}_t = \inf\{k \in \mathbb{Z} : \exists g \in \mathcal{F}_k \text{ s.t. } g(X_i) = f(X_i),\ \forall i \leq t\}$
Let $\widehat{\mathcal{F}}_{I_t} = \{g \in \mathcal{F}_{I_t} : g(X_i) = f(X_i),\ \forall i \leq t\}$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure 1.5: Adaptive algorithm over the nested sequence $\mathcal{F}_k$

indexed by a non-decreasing sequence $k_{i\in\mathbb{Z}}$ of Lipschitz constants defining a mesh grid of $\mathbb{R}^+$ (*i.e.* such that $\forall x \in \mathbb{R}$, $\exists i \in \mathbb{Z}$ with $k_i \leq x < k_{i+1}$). For these strategies, we derive consistency results, finite-time bounds on the difference between the maximum and its approximation, as well as fast rates under an additional Hölder like condition. A summary of these results can be found in the Section 1.4.3.

Application to non-smooth functions. The third chapter of the thesis is dedicated to the analysis and implementation of the generic optimization scheme to non-smooth functions. In this setting, the first algorithm is implemented with a set $\mathcal{F}$ set to $\{f : \mathcal{X} \to \mathbb{R} \mid r_f \in \mathcal{R}\}$ where $r_f : (x, x') \mapsto \operatorname{sgn}(f(x) - f(x'))$ denotes the ranking rule induced by f and $\mathcal{R}$ is a fixed set of ranking rules set as input. The adaptive algorithm performs model selection over the sequence $\{f : \mathcal{X} \to \mathbb{R} \mid r_f \in \mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ indexed by a nested sequence of sets of ranking rules:

$$\mathcal{R}_1 \subseteq \mathcal{R}_2 \subseteq \cdots \subseteq \mathcal{R}_\infty.$$

For these strategies, we identify sufficient conditions for consistency and derive finite-bounds on the Euclidean distance between the maximum and its approximation under a condition controlling the regularity of the level sets of the function. Additionally, several equivalences between various optimization problems are also provided at the end of the chapter as they support the implementation of the algorithm for specific ranking structures. A summary of these results can be found in the Section 1.4.4.

1.4.2 Limitations

Before presenting a summary of the results obtained for these algorithms, we first discuss some of their limitations.

Practical implementation. Due to the theoretical nature and the genericity of the algorithms, several questions remain to be addressed in order to derive practical implementations. In particular, the crucial steps of (i) identifying the set of functions which are consistent with the sample, and (ii) testing whether there exists a function consistent with the sample that can return an evaluation point at least equal to the best evaluation observed so far might not be simple in practice. Nevertheless, we point out that for specific sets of functions a complete implementation for both the algorithms can be proposed (see Chapter 3 and Section 3.5 for further discussions on this topic).

Extension to noisy evaluations. As can be seen, the algorithms can only deal with noiseless evaluations since they were not tailored to handle the noisy scenario. However, we point out that several extensions are proposed in Chapters 2 and 3, Sections 2.3 and 3.5 to extend the algorithms to settings with noisy evaluations depending on the set of functions $\mathcal{F}$. In particular, given a set $\mathcal{F}$ and a sample $(X_1,Y_1),\dots,(X_t,Y_t)$ of $t \geq 1$ noisy evaluations where $Y_i = f(X_i) + \sigma\epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0,1)$, a plausible strategy would consist in using a relaxed version of the active subset of consistent function $\mathcal{F}_{\delta,t} = \{g \in \mathcal{F} : R_t(g) \leq \min_{g\in\mathcal{F}} R_t(g) + UB_{\delta,t}\}$ where $R_t(g) = (1/t)\sum_{i=1}^t (f(X_i) - g(X_i))^2$ denotes, for instance, the empirical mean squared error and the term $UB_{\delta,t}$ comes out of some generalization bounds on the deviation $|R_t(f) - \min_{g\in\mathcal{F}} R_t(g)|$ which depend on the set of functions $\mathcal{F}$.

Rejection method in high dimension. Finally, it should be noticed that using the rejection method to sample the next evaluation points can be computationally disuasive in the case of high dimensional input spaces. Indeed, a simple union bound indicates that the complexity of generating the next evaluation point $X_{t+1} \mid \{X_i\}_{i=1}^t \sim \mathcal{X}_t$ given a sample $\{(X_i, f(X_i))\}_{i=1}^t$ over the set of points $\mathcal{X}_t := \{x \in \mathcal{X} : \exists g \in \mathcal{F}_t \text{ s.t. } g(x) \geq \max_{i=1\dots t} f(X_i)\}$ is lower bounded, with probability at least $1-\delta$, by the complexity of testing whether there exists a function in $g \in \mathcal{F}_t$ satisfying the decision rule multiplied by $\lfloor \delta \cdot \mu(\mathcal{X})/\mu(\mathcal{X}_t) \rfloor$. In particular, recalling that the volume $\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\})$ of the ϵ-level set of a k-Lipschtz functions can easily be of order ϵ^d (*e.g.* the sphere function), we deduce that the numerical complexity of reaching the maximum with precision ϵ can be at least of order $\Omega_{\mathbb{P}}(e^{d\cdot\ln(1/\epsilon)})$ and thus exponentially increasing with the dimensionality d. However, we point out that techniques using Markov chain sampling such as the one proposed in Reaume et al. (2001) could be considered in order to reduce the computational time.

1.4.3 Optimization of Lipschitz functions

In this section, we provide a summary of the main theoretical results that have been obtained in smooth optimization.

Introduction. Chapter 2 focuses on the smoothness-based approach to global optimization. This approach is based on the simple observation that, in many applications,

the function presents some regularity with regards to the input. However, except from the recent works of Munos (2014) and Grill et al. (2015) which consider local assumptions, very few is known about the convergence properties of optimization algorithms that exploit the global smoothness of the function. The analysis provided in this chapter considers that the target function has finite Lipschitz constant, *i.e.*, $\exists k \geq 0$ s.t. $|f(x) - f(x')| \leq k \cdot \|x - x'\|_2$ for all $(x, x') \in \mathcal{X}^2$. Our contribution is twofold. First, we introduce a novel algorithm that requires the knowledge of the Lipschitz constant and its adaptive version which estimates the constant. Second, we provide convergence results in terms of consistency and finite-time bounds on the difference between the maximum and its approximation.

Generic results. To assess optimality, we start by casting two generic results about the convergence of sequential algorithms over the class of Lipschitz functions. The first result is a necessary and sufficient condition for consistency: a sequential algorithm is consistent over the set of Lipschitz functions if and only if

$$\forall f \in \bigcup_{k \geq 0} \mathrm{Lip}(k), \ \sup_{x \in \mathcal{X}} \min_{i=1\ldots n} \|X_i - x\|_2 \xrightarrow{p} 0$$

where $X_1, \ldots, X_n$ denotes a sequence of n evaluation points generated by the algorithm over f. The second result is a minimax rate of convergence on the set of Lipschitz functions with fixed constant: for any $n \in \mathbb{N}^\star$ and any $k \geq 0$, we have

$$c_1 \cdot k \cdot n^{-\frac{1}{d}} \leq \inf_{A \in \mathcal{A}} \sup_{f \in \mathrm{Lip}(k)} \mathbb{E}\left[\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i)\right] \leq c_2 \cdot k \cdot n^{-\frac{1}{d}}$$

where $\mathcal{A}$ denotes the set of sequential strategies, c_1 and c_2 are two constants which only depend on $\mathcal{X}$ and $X_1, \ldots, X_n$ denotes a sequence of n evaluation points generated by A over f.

Optimization with fixed Lipschitz constant. The LIPO algorithm (displayed in Figure 2.1, Chapter 2) aims at optimizing any k-Lipschitz function for a given $k \geq 0$ set as input. For this strategy, we provide a first finite-time bound on the difference between the maximum and its approximation: for any $f \in \mathrm{Lip}(k)$, any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, we have with probability at least $1 - \delta$,

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d}}$$

where $X_1, \ldots, X_n$ denotes a sequence of n evaluation points generated by LIPO over f. This bound proves that LIPO is consistent and achieves minimax efficiency over the set of k-Lipschitz functions. One can also derive fast convergence rates under an additional assumption: let f be any k-Lipschitz with a unique optimizer $x^\star \in \mathcal{X}$ and such that there exists $\kappa \geq 1$ and $c_\kappa > 0$ for which we have $f(x^\star) - f(x) \geq c_\kappa \cdot \|x^\star - x\|_2^\kappa$ for all $x \in \mathcal{X}$, then, for any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, we have with probability at least $1 - \delta$,

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \times \begin{cases} c_1 \left(\frac{\ln(n/\delta)}{n}\right)^{\frac{\kappa}{d(\kappa-1)}} & \text{if } \kappa > 1 \\ e^{-c_2 n / \ln(n/\delta)} & \text{if } \kappa = 1. \end{cases}$$

where c_1 and c_2 are two constants depending on $\mathcal{X}$, k, κ and c_κ. A comparison of these results to existing works can be found in Chapter 2, Sections 2.3.4 and 2.4.3.

Optimization with unknown Lipschitz constant. The AdaLIPO algorithm (displayed in Figure 2.4, Chapter 2) is an adaptive version of LIPO which takes as input a parameter $p \in (0,1)$ and a sequence of Lipschitz constants $k_{i\in\mathbb{Z}}$ defining a mesh-grid of $\mathbb{R}^+$ (*i.e.* such that for all $x \in \mathbb{R}$ there exists $i \in \mathbb{Z}$ satisfying $k_i \leq x < k_{i+1}$). For this procedure, we derive similar results as for LIPO. The first result is an upper bound on the difference between the true maximum and its approximation: for any $f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$, any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, we have with probability at least $1-\delta$:

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{5}{p} + \frac{2\ln(\delta/3)}{p\ln(1-\Gamma(f,k_{i\in\mathbb{Z}}))}\right)^{\frac{1}{d}} \cdot \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}}$$

where $\Gamma(f,k_{i\in\mathbb{Z}})$ is a positive constant defined in the core of the document which measures the smoothness of f with regards to $k_{i\in\mathbb{Z}}$ and $X_1,\dots,X_n$ denotes a sequence of n evaluation points generated by the AdaLIPO over f. Again, one can derive tighter convergence results under an additional assumption: for any function $f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$ which has a unique optimizer $x^\star \in \mathcal{X}$ and such that there exists $\kappa \geq 1$ and $c_\kappa > 0$ for which we have $f(x^\star) - f(x) \geq c_\kappa \cdot \|x^\star - x\|_2^\kappa$ for all $x \in \mathcal{X}$, then, for any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, we have with probability at least $1-\delta$:

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot C_{f,p,\delta} \times \begin{cases} c_1 \left(\frac{\ln(n/\delta)}{n}\right)^{\frac{\kappa}{d(\kappa-1)}} & \text{if } \kappa > 1 \\ e^{-c_2 n/\ln(n/\delta)} & \text{if } \kappa = 1. \end{cases}$$

where c_1 and c_2 are two constants depending on $\mathcal{X}$, $k_{i\in\mathbb{Z}}$, κ and c_κ and $C_{f,p,\delta}$ depends on $f, k_{i\in\mathbb{Z}}, \delta$ and p. A discussion on these results can be found in the core of the document.

Future directions. Possible future research directions include (i) deriving problem-dependent lower bounds, (ii) refining the bounds of Theorem 2.18 in order to match the bounds of Theorem 2.17, (iii) extending the algorithms to settings with noisy evaluations and (iv) investigating better values for the choice of p as well as for the sequence of Lipschitz constants $k_{i\in\mathbb{Z}}$. Finally, two open questions would be to determine (v) when do the algorithms exploiting the global smoothness of the function perform better than the one only exploiting the local smoothness and (vi) how do they scale to the dimensionality of the input space.

1.4.4 Non-smooth optimization and ranking

In this section, we provide a summary of the main theoretical results that have been obtained in non-smooth optimization.

Introduction. In Chapter 3, we propose to explore concepts from ranking theory based on overlaying estimated level sets (Clémençon and Vayatis (2010)) in order to develop global optimization methods that do not rely on the smoothness of the function. The idea behind this approach is simple: even if the unknown function presents arbitrary large variations, most of the information required to identify its optimum may be contained in

its induced ranking rule, i.e. how the level sets of the function are included one in another. To exploit this idea, we introduce a novel optimization scheme where the complexity of the function is characterized by the underlying pairwise ranking that it defines. Our contribution is twofold: first, we introduce two novel global optimization algorithms that learn the ranking rule induced by the unknown function, and second, we provide mathematical results in terms of statistical consistency and convergence to the optimum.

Ranking structure. As a first go, we introduce the key concepts that are at the core of our strategy. We define the ranking rule induced by a function and deduce a notion of complexity for optimization through ranking structures. The ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1,0,1\}$ induced by a function $f : \mathcal{X} \to \mathbb{R}$ is defined by:

$$r_f(x,x') = \begin{cases} +1 & \text{if } f(x) > f(x') \\ 0 & \text{if } f(x) = f(x') \\ -1 & \text{if } f(x) < f(x') \end{cases}$$

for all $(x,x') \in \mathcal{X}^2$. The key argument of our approach is that even if the function f might be complex or present arbitrary large variations, its induced ranknig rule may be simple. We thus introduce the notion of ranking structure, a collection of ranking rules, to define the complexity of r_f. In addition to the very large continuous ranking structure $\mathcal{R}_\infty := \{r_g : g \in \mathcal{C}^0(\mathcal{X})\}$, we provide some examples of more stringent ranking structures such as the set of polynomial ranking rules of degree $k \geq 1$ defined by:

$$\mathcal{R}_{\mathcal{P}_k} := \{r_h : (x,x') \mapsto \operatorname{sgn}(h(x) - h(x')) \mid h \in \mathcal{P}_k(\mathcal{X},\mathbb{R})\}.$$

In this setting, the index $k^\star = \min\{k \in \mathbb{N}^\star : r_f \in \mathcal{R}_{\mathcal{P}_k}\}$ of the smallest ranking structure containing the ranking rule r_f induced by the target function defines its complexity.

Optimization with fixed ranking structure. The RankOpt algorithm (displayed in Figure 3.3, Chapter 3) aims at optimizing any function f which induces a ranking rule in a fixed ranking structure $\mathcal{R}$ set as input. For this algorithm, we identify a sufficient condition for consistency: if the function f induces a ranking rule r_f in $\mathcal{R}$ and has an identifiable maximum (*i.e.* $\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}) > 0$ for all $\epsilon > 0$), then, we have

$$\max_{i=1\ldots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x)$$

where $X_1, \ldots X_n$ denotes a sequence of n evaluation points generated by RankOpt over f. By considering an additional condition which captures the regularity of the level sets of the function around its maximum, one can derive a nonasymptotic result: for any function f inducing a ranking rule r_f in $\mathcal{R}$ which has a unique optimizer and (c_α,α)-regular level sets (i.e. $\exists c_\alpha > 0$ and $\alpha \geq 0$ for which we have $\inf_{x\in f^{-1}(y)} \|x^\star - x\| \leq c_\alpha \cdot \sup_{x\in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)}$ for all $y \in \operatorname{Im}(f)$), then, for all $n \in \mathbb{N}$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$:

$$\|x^\star - X_{\hat{\imath}_n}\|_2 \leq C_\alpha \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$

where C_α is a constant which only depends on α and $\mathcal{X}$. Additionally to these results, an exponentially decreasing lower bound is also provided in the core of the document.

Adaptive algorithm. The AdaRankOpt algorithm (displayed in Figure 3.4, Chapter 3) is an adaptive version of the RankOpt algorithm which uses a parameter $p \in (0,1)$ and a nested sequences of nested ranking structure: $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \cdots \subset \mathcal{R}_\infty$. For this algorithm, one can derive a finite-time bound by using a the Rademacher complexity of a ranking structure $\mathcal{R}$ with regards to a ranking rule r_f introduced in Clémençon et al. (2010) and defined by:

$$R_n(\mathcal{R}) := \mathbb{E}\left[\sup_{r\in\mathcal{R}} \frac{1}{\lfloor n/2\rfloor}\left|\sum_{i=1}^{\lfloor n/2\rfloor} \epsilon_i \mathbb{I}\left\{r(X_i, X_{i+\lfloor n/2\rfloor}) \neq r_f(X_i, X_{i+\lfloor n/2\rfloor})\right\}\right|\right]$$

where $\epsilon_1,\ldots,\epsilon_{\lfloor n/2\rfloor}$ are $\lfloor n/2\rfloor$ rademacher variables (random sign variables) and $X_1\ldots,X_n$ are n i.i.d. random variables uniformly distributed over $\mathcal{X}$. More specifically, assuming that (i) the index $N^\star = \min\{k \in \mathbb{N}^\star : r_f \in \mathcal{R}_k\}$ of the smallest ranking structure of the sequence containing r_f is finite, (ii) there exists $K > 0$ such that $R_n(\mathcal{R}_{N^\star}) \leq \sqrt{K/n}$ for all $n > 0$ and (iii) the function has a unique optimizer and (c_α, α)-regular level sets, then, for any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, we have with probability at least $1-\delta$:

$$\|x^\star - X_{\hat{\imath}_n}\| \leq C_\alpha \cdot \left(\frac{11(K+\ln(4/\delta))}{p \inf_{r\in\mathcal{R}_{N^\star-1}} L(r)^2}\right) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$

where $X_{\hat{\imath}_n}$ denotes the output of the AdaRankOpt algorithm after n iterations and $L(r) = \mathbb{P}(r(X,X') \neq r_f(X,X'))$ denotes the ranking loss where $(X,X') \sim \mathcal{U}(\mathcal{X}\times\mathcal{X})$. In addition to these bounds, a collection of results supporting the implementation of the algorithm for specific ranking structures are provided in the core of the document.

Future directions. Possible future research directions include (i) refining the characterization of a function with regards to a ranking structure in order to identify the class of functions providing the exponentially decreasing loss obtained in the lower bound, (ii) determining better values for the exploration/exploitation parameter p as well as for the choice of sequence of ranking structures set as input and (iii) extending the algorithms to settings with noisy evaluations. As detailed at the end of the chapter, another open perspective would be to develop and analyze a Bayesian version of the algorithm which uses a prior distribution on the ranking rule induced by the unknown function.

1.4.5 Numerical experiments

Introduction. The last chapter of the thesis is dedicated to an empirical comparison of nine global optimization implementations. We compare the AdaLIPO and AdaRank algorithms with seven state-of-the-art global optimization algorithms over a benchmark including real and synthetic optimization problems. The algorithms are compared according to various criteria, including their ability to find a near-optimal solution for various accuracies. As a result, we find that both the algorithms developed in the thesis display competitive results for medium accuracy over most problems of the benchmark but could certainly be improved to greater precision. Eventually, we

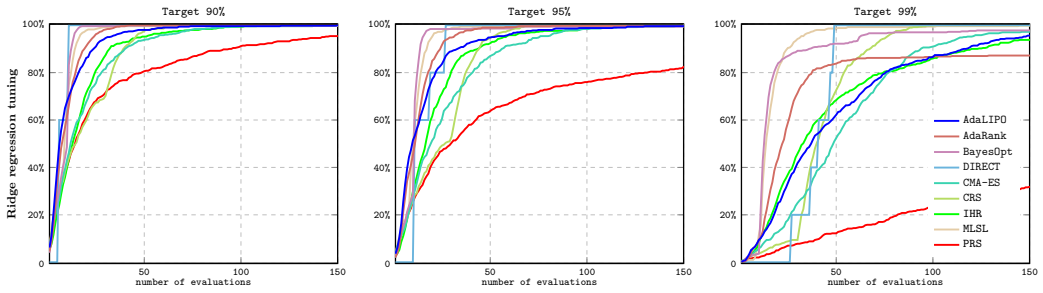


Figure 1.6: Proportion of runs that reached the targets 90%, 95% and 99% in terms of function evaluations over each series of optimization problems.

propose several extensions that might improve their performance at the end of the chapter.

Protocol. The AdaRankOpt (Chapter 2) and AdaLIPO (Chapter 3) algorithms are compared with seven algorithms developed from different approaches of global optimization: BayesOpt (bayesian method, Martinez-Cantin (2014)), CMA-ES (genetic algorithm, Hansen (2006)), CRS (variation of the random search which includes local mutations, Kaelo and Ali (2006)), DIRECT (partitioning technique, Jones et al. (1993)), IHR (generation of a random direction at each step, Zabinsky et al. (1993)), MLSL (series of local optimizations, Kan and Timmer (1987)) et PRS (Pure Random Search, Brooks (1958)). The test bed includes two series of hyperparameter tuning tasks (i.e. ridge regression and neural nets tuning) and two series of synthetic functions that are commonly met in optimization benchmarks. For each algorithm A, objective function f and target parameter t =90%, 95% and 99%, we performed $K = 100$ distinct runs with a total budget of $n = 1000$ evaluations and we have collected the stopping times corresponding to the number of evaluations required by each method to reach the specified target:

$$\tau(A,f,k,t) = \min\{i = 1,\dots,n : f(X_i^{(k)}) \geq f_{\text{target}}(t)\}$$

where $f(X_i^{(k)}),\dots,f(X_n^{(k)})$ denotes a sequence of n evaluations made by the algorithm A on the k-th run and $f_{\text{target}}(t)$ is a precision parameter defined in the core of the document. Based on these stopping times, we measured performance through a collection of indicators, such as the proportion of runs that reached the target in terms of function evaluations or the average number of evaluations required to reach the target:

$$\forall i \leq n,\ \widehat{\mathbb{P}}(\tau < i) := \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\{\tau(A,f,k,t) < i\}, \quad \widehat{\tau}_K := \frac{1}{K}\sum_{k=1}^{K} \tau(A,f,k,t).$$

Results. The performance of the algorithms were aggregated and compared over each series of experiments. We make several comments. First, we find out that none of the algorithm performs uniformly better than the others on each problem of the benchmark. However, in terms of speed of convergence, the MLSL algorithm obtains several times the best results over the regression tuning problems and the first series of synthetic functions while DIRECT and CRS obtain several times the best results over the first series of synthetic functions and the neural nets tuning problems. In addition, we also

find out that the BayesOpt and MLSL algorithms need in average less evaluations than CMA-ES, CRS and IHR to reach a near-optimal solution while AdaLIPO and AdaRank generally rank between the latter (see, e.g., Figure 1.6). Finally, we identify some limits of the algorithms (for instance, AdaLIPO and AdaRank can sometimes reach the 95% target with few evaluations while getting more slowly to the 99% target) and we propose some extensions that might improve their performance.

Future directions. Several directions are discussed at the end of the chapter to improve the performance of the methods. It includes (i) adding a noise parameter to improve their stability, (ii) using near-random covering methods instead of the Pure Random Search, (iii) incorporating local searches during the optimization process and (iv) investigating a dynamic exploration/exploitation trade-off as well as better parameters for the model selection. Finally, a last open question would be to determine whether using Bayesian models within these methods could lead to better empirical results without deteriorating their convergence properties.

2

Global optimization of Lipschitz functions

Abstract. The goal of this chapter is to design sequential strategies which lead to efficient optimization of an unknown function under the only assumption that it has a finite Lipschitz constant. We first identify sufficient conditions for the consistency of generic sequential algorithms and formulate the expected minimax rate for their performance. We introduce and analyze a first algorithm called LIPO which assumes the Lipschitz constant to be known. Consistency, minimax rates for LIPO are proved, as well as fast rates under an additional Hölder like condition. An adaptive version of LIPO is also introduced for the more realistic setup where the Lipschitz constant is unknown and has to be estimated along with the optimization. Eventually, similar theoretical guarantees are shown to hold for the adaptive LIPO algorithm. A significant part of this work has been published in Malherbe and Vayatis (2017).

Contents

2.1 Introduction

In this chapter, we focus on the smoothness-based approach to global optimization. This approach is based on the simple observation that, in many applications, the system presents some regularity with respects to the input. In particular, the use of the Lipschitz constant, first proposed in the seminal works of Shubert (1972) and Piyavskii (1972), initiated an active line of research and played a major role in the development of many efficient global optimization algorithms such as DIRECT (Jones et al. (1993)), MCS (Huyer and Neumaier (1999)) or more recently SOO (Preux et al. (2014)). Convergence properties of global optimization methods have been developed in the works of Valko et al. (2013) and Munos (2014) under local smoothness assumptions, but, up to our knowledge, such properties have not been considered in the case where only the global smoothness of the function can be specified. An interesting question is how much global assumptions on regularity which cover in some sense local assumptions may improve the convergence of the latter. In this work, we address the following questions: (i) find the limitations and the best performance that can be achieved by any algorithm over the class of Lipschitz functions and (ii) design efficient and optimal algorithms for this class of problems. Our contribution with regards to the above mentioned works is twofold. First, we introduce two novel algorithms for global optimization which exploit the global smoothness of the unknown function and display good performance in typical benchmarks for optimization. Second, we show that these algorithms achieve faster rates of convergence on globally smooth problems than the previously known methods which only exploit the local smoothness of the function. The rest of the chapter is organized as follows. In Section 2.2, we recall the framework and provide generic results about the convergence of global optimization algorithms. In Section 2.3, we introduce and analyze the LIPO algorithm which requires the knowledge of the Lipschitz constant. Finally, the algorithm is extended to the case where the Lipschitz constant is not assumed to be previously known in Section 2.4. All proofs are postponed to Section 2.6 and an emprical comparison of the adaptive version of the algorithm to existing methods can be found in Chapter 4.

2.2 Setup and preliminary results

2.2.1 Setup and notations

Setup. Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact and convex set with non-empty interior and let $f : \mathcal{X} \to \mathbb{R}$ be an unknown function which is only supposed to admit a maximum over its input domain. The goal in global optimization consists in finding some point

$$x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$$

with a minimal amount of function evaluations. The standard setup involves a sequential procedure which starts by evaluating the function at a random initial point $X_1 \in \mathcal{X}$ and then iteratively selects at each step $t \geq 1$ a random evaluation point $X_{t+1} \in \mathcal{X}$ which depends on the previous evaluations $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ and is $\mathcal{F}_t$-measurable where $\mathcal{F}_t = \sigma((X_1, f(X_1)), \ldots, (X_t, f(X_t)))$ is the filtration generated by the history of available information, and recieves the evaluation of the unknown function $f(X_{t+1})$ at this point. After n iterations, we consider that the algorithm returns an evaluation point

$X_{\hat{\imath}_n}$ with $\hat{\imath}_n \in \arg\min_{i=1\dots n} f(X_i)$ which has recorded the highest evaluation. The performance of the algorithm over the function f is then measured after n iterations through the difference between the value of the true maximum and the value of the highest evaluation observed so far:

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i).$$

The analysis provided in the paper considers that the number n of evaluation points is not fixed and it is assumed that function evaluations are noiseless. Lastly, the assumption made on the unknown function f throughout the paper is that it has a finite Lipschitz constant k, *i.e.*

$$\exists k \geq 0 \text{ s.t. } |f(x) - f(x')| \leq k \cdot \left\|x - x'\right\|_2 \ \forall (x,x') \in \mathcal{X}^2.$$

Before starting the analysis, we point out that similar settings have also been studied in the works of Munos (2014) and Malherbe and Vayatis (2016) and that Valko et al. (2013) and Grill et al. (2015) considered the noisy scenario.

Notations. For all $x = (x_1,\dots,x_d) \in \mathbb{R}^d$, we denote by $\|x\|_2^2 = \sum_{i=1}^d x_i^2$ the standard ℓ_2-norm and by $B(x,r) = \{x' \in \mathbb{R}^d : \|x - x'\|_2 \leq r\}$ the ball centered in x of radius $r \geq 0$. For any bounded set $\mathcal{X} \subset \mathbb{R}^d$, we define its inner-radius as $\mathrm{rad}(\mathcal{X}) = \max\{r > 0 : \exists x \in \mathcal{X} \text{ such that } B(x,r) \subseteq \mathcal{X}\}$, its diameter as $\mathrm{diam}(\mathcal{X}) = \max_{(x,x')\in\mathcal{X}^2} \|x - x'\|_2$ and we denote by $\mu(\mathcal{X})$ its volume where $\mu(\cdot)$ stands for the Lebesgue measure. Additionally, $\mathrm{Lip}(k) := \{f : \mathcal{X} \to \mathbb{R} \text{ s.t. } |f(x) - f(x')| \leq k \cdot \|x - x'\|_2,\ \forall (x,x') \in \mathcal{X}^2\}$ denotes the class of k-Lipschitz functions defined on $\mathcal{X}$ and $\bigcup_{k\geq 0}\mathrm{Lip}(k)$ denotes the set of Lipschitz continuous functions. Finally, $\mathcal{U}(\mathcal{X})$ stands for the uniform distribution over a bounded measurable domain $\mathcal{X}$, $\mathcal{B}(p)$ for the Bernoulli distribution of parameter p, $\mathbb{I}\{\cdot\}$ for the standard indicator function taking values in $\{0,1\}$ and the notation $X \sim \mathcal{P}$ means that the random variable X has the distribution $\mathcal{P}$.

2.2.2 Preliminary results

In order to design efficient procedures, we start to investigate the best performance that can be achieved by any algorithm over the class of Lipschitz functions.

Sequential algorithms and optimization consistency. We first describe the sequential procedures that are considered here and the corresponding concept of consistency in the sense of global optimization.

Definition 2.1. (Sequential algorithm) *The class of optimization algorithms we consider, denoted in the sequel by $\mathcal{A}$, contains all the algorithms $A = \{A_t\}_{t\geq 1}$ completely described by:*

1. *A distribution A_1 taking values in $\mathcal{X}$ which allows to generate the first evaluation point, i.e. $X_1 \sim A_1$;*
2. *An infinite collection of parametric distributions $\{A_t\}_{t\geq 2}$ taking values in $\mathcal{X}$ and based on the previous evaluations which define the iteration loop, i.e. $X_{t+1}|X_1\dots,X_t \sim A_{t+1}((X_1,f(X_1)),\dots,(X_t,f(X_t)))$.*

Note that this class of algorithms also includes deterministic methods in which case the distributions $\{A_t\}_{t\geq 1}$ are degenerate. The next definition introduces the notion of asymptotic convergence.

Definition 2.2. (Optimization Consistency) *A global optimization algorithm A is said to be consistent over a set $\mathcal{F}$ of real-valued functions admitting a maximum over their input domain $\mathcal{X}$ if and only if*

$$\forall f \in \mathcal{F},\ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x)$$

where $X_1,\dots,X_n$ denotes a sequence of n evaluations points generated by the algorithm A over the function f.

Asymptotic performance. We now investigate the minimal conditions for a sequential algorithm to achieve asymptotic convergence. Of course, it is expected that a global optimization algorithm should be consistent at least for the class of Lipschitz functions and the following result reveals a necessary and sufficient condition (NSC) in this case.

Proposition 2.3. (Consistency NSC) *A global optimization algorithm A is consistent over the set of Lipschitz functions if and only if*

$$\forall f \in \textstyle\bigcup_{k\geq 0} \mathrm{Lip}(k),\ \sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|X_i - x\|_2 \xrightarrow{p} 0.$$

A crucial consequence of the latter proposition is that the design of any consistent method ends up to covering the whole input space regardless of the function values. The example below introduces the most popular space-filling method which will play a central role in our analysis.

Example 2.4. (Pure Random Search) *The Pure Random Search* (PRS) *consists in sequentially evaluating the function over a sequence of points $X_1, X_2, X_3, \dots$ uniformly and independently distributed over the input space $\mathcal{X}$. For this method, a simple union bound indicates that for all $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have independently of the function values and with probability at least $1-\delta$,*

$$\sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|X_i - x\|_2 \leq \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{\ln(n/\delta) + d\ln(d)}{n}\right)^{\frac{1}{d}}.$$

In addition to this result, we point out that the covering rate of any method can easily be shown to be at best of order $\Omega(n^{-1/d})$ and thus subject to to the curse of dimensionality by means of covering arguments. Keeping in mind the equivalence of Proposition 2.3, we may now turn to the nonasymptotic analysis.

Finite-time performance. We investigate here the best performance that can be achieved by any algorithm with a finite number of function evaluations. We start by casting a negative result stating that any algorithm can suffer, at any time, an arbitrarily large loss over the class of Lipschitz functions.

Proposition 2.5. *Consider any global optimization algorithm A. Then, for any constant $C > 0$ arbitrarily large, any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, there exists a function $\tilde{f} \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$ only depending on (A, C, n, δ) for which we have with probability at least $1-\delta$,*

$$C \leq \max_{x\in\mathcal{X}} \tilde{f}(x) - \max_{i=1\dots n} \tilde{f}(X_i).$$

This result might not be very surprising since the class of Lipschitz functions includes functions with finite, but arbitrarily large variations. When considering the subclass of functions with fixed Lipschitz constant, it becomes possible to derive finite-time bounds on the minimax rate of convergence.

Proposition 2.6. (Minimax rate), *adapted from Bull (2011). For any Lipschitz constant $k \geq 0$ and any $n \in \mathbb{N}^\star$, the following inequalities hold true:*

$$c_1 \cdot k \cdot n^{-\frac{1}{d}} \leq \inf_{A\in\mathcal{A}} \sup_{f\in\mathrm{Lip}(k)} \mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)\right] \leq c_2 \cdot k \cdot n^{-\frac{1}{d}}$$

where $c_1 = \mathrm{rad}(\mathcal{X})\,/\,(8\sqrt{d})$, $c_2 = \mathrm{diam}(\mathcal{X}) \times d!$ and the expectation is taken over a sequence $X_1, \dots, X_n$ of n evaluation points generated by the algorithm A over f.

We stress that this minimax convergence rate of order $\Theta(n^{-1/d})$ can still be achieved by any method with an optimal covering rate of order $O(n^{-1/d})$. Observe indeed that since $\mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)\right] \leq k \times \mathbb{E}\left[\sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|x - X_i\|_2\right]$ for all $f \in \mathrm{Lip}(k)$, then an optimal covering rate necessarily implies minimax efficiency. However, as it can be seen by examining the proof of Proposition 2.6 provided in the Appendix Section, the functions constructed to prove the limiting bound of $\Omega(n^{-1/d})$ are spikes which are almost constant everywhere and do not present a large interest from a practical perspective. In particular, we will see in the sequel that one can design:

I) An algorithm with fixed constant $k \geq 0$ which achieves minimax efficiency and also presents exponentially decreasing rates over a large subset of functions, as opposed to space-filling methods (LIPO, Section 2.3).

II) A consistent algorithm which does not require the knowledge of the Lipschitz constant and presents comparable performance as when the constant k is assumed to be known (AdaLIPO, Section 2.4).

2.3 Optimization with fixed Lipschitz constant

In this section, we consider the problem of optimizing an unknown function f given the knowledge that $f \in \mathrm{Lip}(k)$ for a given $k \geq 0$.

2.3.1 The LIPO Algorithm

The inputs of the LIPO algorithm (displayed in Figure 2.1) are a number n of function evaluations, a Lipschitz constant $k \geq 0$, the input space $\mathcal{X}$ and the unknown function $f \in \mathrm{Lip}(k)$. At each iteration $t \geq 1$, a random variable X_{t+1} is sampled uniformly over the input space $\mathcal{X}$ and the algorithm decides whether or not to evaluate the function at this point. Indeed, it evaluates the function over X_{t+1} if and only if the value of the

Input: $n \in \mathbb{N}^\star$, $k \geq 0$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \text{Lip}(k)$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $\min_{i=1\dots t}(f(X_i) + k \cdot \|X_{t+1} - X_i\|_2) \geq \max_{i=1\dots t} f(X_i)$ {Decision rule}
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure 2.1: The LIPO algorithm

upper bound on possible values $UB_{k,t} : x \mapsto \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2$ evaluated at this point and computed from the previous evaluations, is at least equal to the value of the best evaluation observed so far $\max_{i=1\dots t} f(X_i)$. To illustrate how the decision rule operates in practice, an example of the computation of the upper bound can be found in Figure 2.2.

More formally, the mechanism behind the decision rule can be explained using the active subset of consistent functions previously considered in active learning (see, *e.g.*, Dasgupta (2011) or Hanneke (2011)).

Definition 2.7. (Consistent functions) *The active subset of k-Lipschitz functions consistent with the unknown function f over a sample $(X_1, f(X_1)), \dots, (X_t, f(X_t))$ of $t \geq 1$ evaluations is defined as follows:*

$$\mathcal{F}_{k,t} := \{g \in \text{Lip}(k) : \forall i \in \{1 \dots t\},\ g(X_i) = f(X_i)\}.$$

One can indeed recover from this definition the subset of points which can actually maximize the target function f.

Definition 2.8. (Potential maximizers) *Using the same notations as in Definition 2.7, we define the subset of potential maximizers estimated over any sample $t \geq 1$ evaluations with a constant $k \geq 0$ as follows:*

$$\mathcal{X}_{k,t} := \left\{ x \in \mathcal{X} : \exists g \in \mathcal{F}_{k,t} \text{ such that } x \in \arg\max_{x \in \mathcal{X}} g(x) \right\}.$$

In the continuation of this definition, we may now provide an equivalence which makes the link with the decision rule of the LIPO algorithm.

Lemma 2.9. *If $\mathcal{X}_{k,t}$ denotes the set of potential maximizers defined above, then we have the following equivalence:*

$$x \in \mathcal{X}_{k,t} \Leftrightarrow \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2 \geq \max_{i=1\dots t} f(X_i).$$

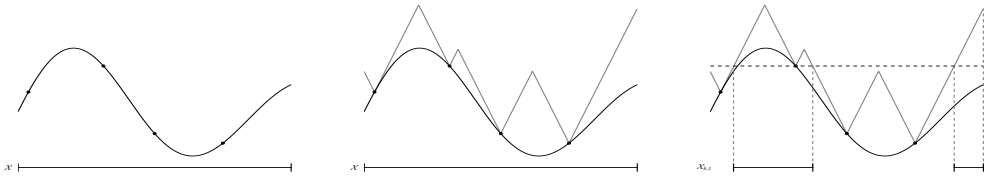


Figure 2.2: *Left:* A Lipschitz function and a sample of $t = 4$ evaluations. *Middle:* In grey, the upper bound $UB : x \mapsto \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2$. *Right:* the set of points $\mathcal{X}_{k,t} := \{x \in \mathcal{X} : UB(x) \geq \max_{i=1\dots t} f(X_i)\}$ which satisfy the decision rule.

Thus, we deduce from this lemma that the LIPO algorithm only evaluates the function over points that still have a chance to be a maximizer of the unknown function.

Remark 2.10. (Adaptation to noisy evaluations) *It is noteworthy that the LIPO algorithm could easily be extended to settings with noisy evaluations by slightly adapting the ideas developed in Dasgupta (2011) and Hanneke (2011). More specifically, considering a sample $(X_1, Y_1), \dots, (X_n, Y_n)$ of $n \geq 1$ noisy observations where $Y_i = f(X_i) + \sigma\epsilon_i$ and $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0,1)$ and observing that the empirical mean-squared error $R_n(f) := (1/n)\sum_{i=1}^n (f(X_i) - Y_i)^2 = (\sigma^2/n)\sum_{i=1}^n \epsilon_i^2$ evaluated in f is distributed as a chi-square, a possible approach would simply consist in using a relaxed version of the active subset $\mathcal{F}_{k,\delta,t} := \{g \in \mathrm{Lip}(k) : R_n(g) \leq (\sigma^2/n) \cdot \chi^2_{1-\delta,n}\}$ where $\chi^2_{1-\delta,n}$ denotes the $1-\delta$ quantile of the chi-squared distribution with n degrees of freedom.*

Remark 2.11. (Extension to other smoothness assumptions) *It is also important to note the proposed optimization scheme could easily be extended to a large number of classes of globally and locally smooth functions by slightly adapting the decision rule. For instance, when $\mathcal{F}_\ell = \{f : \mathcal{X} \to \mathbb{R} \mid x^\star$ is unique and $\forall x \in \mathcal{X}, f(x^\star) - f(x) \leq \ell(x^\star, x)\}$ denotes the set of functions previously considered in Munos (2014) which are locally smooth around their maximum with regards to a given semi-metric $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, a straightforward derivation of Lemma 2.9 directly gives that the decision rule applied in X_{t+1} would simply consists in testing whether $\max_{i=1\dots t} f(X_i) \leq \min_{i=1\dots t} f(X_i) + \ell(X_{t+1}, X_i)$. However, since the purpose of this work is to design fast algorithms for Lipschitz functions, we only derive convergence results for the version of the algorithm stated above.*

2.3.2 Convergence analysis

We start by casting the consistency property of the algorithm.

Proposition 2.12. (Consistency) *For any Lipschitz constant $k \geq 0$, the LIPO algorithm tuned with a parameter k is consistent over the set k-Lipschitz functions, i.e.*

$$\forall f \in \mathrm{Lip}(k), \ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x \in \mathcal{X}} f(x).$$

The next result shows that the value of the highest evaluation observed by the algorithm is always superior or equal in the usual stochastic ordering sense to the one of a Pure Random Search.

Proposition 2.13. (Faster than pure random search) *Consider the LIPO algorithm tuned with any constant $k \geq 0$. Then, for any $f \in \mathrm{Lip}(k)$ and $n \in \mathbb{N}^\star$, we have that $\forall y \in \mathbb{R}$,*

$$\mathbb{P}\left(\max_{i=1\dots n} f(X_i) \geq y\right) \geq \mathbb{P}\left(\max_{i=1\dots n} f(X'_i) \geq y\right)$$

where $X_1, \dots, X_n$ is a sequence of n evaluation points generated by LIPO and $X'_1, \dots, X'_n$ is a sequence of n independent random variables uniformly distributed over $\mathcal{X}$.

Based on this result, one can easily derive a first finite-time bound on the difference between the value of the true maximum and its approximation.

Corollary 2.14. (Upper bound) *For any $f \in \mathrm{Lip}(k)$, any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d}}.$$

This bound which proves the miminax optimality of LIPO stated in Proposition 2.6 once integrated does however not show any improvement over PRS and it cannot be significantly improved without making any additional assumption as shown below.

Proposition 2.15. *For any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, there exists a function $\tilde{f} \in \mathrm{Lip}(k)$, only depending on n and δ, for which we have with probability at least $1-\delta$:*

$$k \cdot \mathrm{rad}(\mathcal{X}) \cdot \left(\frac{\delta}{n}\right)^{\frac{1}{d}} \leq \max_{x\in\mathcal{X}} \tilde{f}(x) - \max_{i=1\dots n} \tilde{f}(X_i).$$

As announced in Section 2.2.2, one can nonetheless get tighter polynomial bounds and even an exponential decay by using the following condition which describes the behavior of the function around its maximum.

Condition 2.16. (Decreasing rate around the maximum) *A function $f : \mathcal{X} \to \mathbb{R}$ is (κ, c_κ)-decreasing around its maximum for some $\kappa \geq 0$, $c_\kappa \geq 0$ if:*

1. *The global optimizer $x^\star \in \mathcal{X}$ is unique;*
2. *For all $x \in \mathcal{X}$, we have that:*

$$f(x^\star) - f(x) \geq c_\kappa \cdot \|x - x^\star\|_2^\kappa.$$

This condition, already considered in the works of Zhigljavsky and Pintér (1991) and Munos (2014), captures how fast the function decreases around its maximum. It can be seen as a local one-sided Hölder condition that can only be met for $\kappa \geq 1$ when f is assumed to be Lipschitz. As an example, three functions satisfying this condition with different values of κ are displayed on Figure 2.5.

Theorem 2.17. (Fast rates) *Let $f \in \mathrm{Lip}(k)$ be any Lipschitz function satisfying Condition 2.16 for some $\kappa \geq 1, c_\kappa > 0$. Then, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq$$

$$k \times \mathrm{diam}(\mathcal{X}) \times \begin{cases} \exp\left\{ - C_{k,\kappa} \cdot \dfrac{n\ln(2)}{\ln(n/\delta) + 2(2\sqrt{d})^d} \right\}, & \kappa = 1, \\ \dfrac{2^\kappa}{2}\left(1 + C_{k,\kappa} \cdot \dfrac{n(2^{d(\kappa-1)} - 1)}{\ln(n/\delta) + 2(2\sqrt{d})^d} \right)^{-\frac{\kappa}{d(\kappa-1)}}, & \kappa > 1, \end{cases}$$

where $C_{k,\kappa} = (c_\kappa \max_{x\in\mathcal{X}} \|x - x^\star\|^{\kappa-1} / 8k)^d$.

We stress that this polynomial bound can be slightly improved in the case where f is locally equivalent to $\|x^\star - x\|_2^\kappa$ (*i.e.*, when $\exists c_\kappa, c_2 > 0$, $c_\kappa \|x^\star - x\|_2^\kappa \leq f(x^\star) - f(x) \leq c_2 \|x^\star - x\|_2^\kappa$) and shown to be of order $O^*_{\mathbb{P}}(n^{-\frac{\kappa\times\kappa}{d(\kappa-1)}})$. The last result we provide states an exponentially decreasing lower bound.

Theorem 2.18. (Lower Bound) *For any $f \in \mathrm{Lip}(k)$ satisfying Condition 2.16 for some $\kappa \geq 1, c_\kappa > 0$ and any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$c_\kappa \cdot \mathrm{rad}(\mathcal{X})^\kappa \cdot e^{-\frac{\kappa}{d}\left(n + \sqrt{2n\ln(1/\delta)} + \ln(1/\delta)\right)} \leq \max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i).$$

The next section provides an explicit derivation of the fast rate on some toy examples and a discussion on these results can be found in Section 2.3.4.

2.3.3 Examples

The next examples consider that the optimization is performed over the hypercube $\mathcal{X} = [-R, R]^d$ for some $R > 0$ and $d \geq 1$.

Sphere function. The sphere function $f(x) = 1 - \|x\|_2$ is the canonical example of Lipschitz function. For this function, the Lipschitz continuity is obtained by applying the triangle inequality: $\forall (x,y) \in \mathcal{X}^2$, considering w.l.o.g. that $\|x\|_2 \geq \|y\|_2$, we have $|f(x) - f(y)| = \|x\|_2 - \|y\|_2 = \|y + x - y\|_2 - \|y\|_2 \leq \|x - y\|_2$. Observing now that $x^\star = \vec{0}$ and $f(x^\star) - f(x) = \|x^\star - x\|_2$ for all $x \in \mathcal{X}$, it is easy to see that Condition 2.16 is satisfied with $\kappa = 1$ and $c_\kappa = 1$. Thus, running LIPO tuned with any $k \geq 1$ would provide an exponentially decreasing rate of order $O^*_{\mathbb{P}}(e^{-n/2(16k\sqrt{d})^d})$.

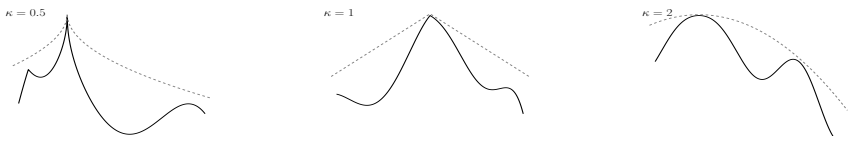


Figure 2.3: Three one-dimensional functions satisfying Condition 2.16 with $\kappa = 1/2$ (*Left*), $\kappa = 1$ (*Middle*) and $\kappa = 2$ (*Right*).

Linear slope. The second class of functions we consider are the linear functions of the form $f(x) = 1 - \langle w, x \rangle$ with weight vectors $w \in \mathbb{R}^d$. In this case, applying Cauchy-Schwartz inequality directly proves the Lipschitz continuity: $\forall (x,y) \in \mathcal{X}^2$, $|f(x) - f(y)| = |\langle w, x - y \rangle| \leq \|w\|_2 \cdot \|x - y\|_2$. Moreover, in the case of non-zero weights, we have that $x^\star = -R \times (\text{sgn}(w_1), \dots, \text{sgn}(w_d))$ and $f(x^\star) - f(x) = \langle w, x - x^\star \rangle \geq \min_{i=1\dots d} |w_i| \cdot \|x^\star - x\|_2$ for all $x \in \mathcal{X}$. Hence, Condition 2.16 is satisfied with $\kappa = 1$ and $c_\kappa = \min_{i=1\dots d} |w_i|$ and we deduce that running LIPO with any $k \geq \|w\|_2$ would provide a decay of order $O^*_{\mathbb{P}}(e^{-n/2(16k\sqrt{d}\min|w_i|)^d})$.

Largest coordinate. The last function we consider is the largest coordinate function $f(x) = 1 - \max\{|x_1|, \dots, |x_d|\}$. Taking any $(x,y) \in \mathcal{X}^2$ and denoting by $i(x)$ and $i(y)$ the indexes of (one of) their largest absolute coordinate we obtain that $|f(x) - f(y)| = |x_{i(x)}| - |y_{i(y)}| \leq |x_{i(x)}| - |y_{i(x)}| \leq |x_{i(x)} - y_{i(x)}| \leq \|x - y\|_2$ by considering w.l.o.g. that $x_{i(x)} \geq y_{i(y)}$. Now, noticing that $x^\star = \vec{0}$ and that $f(x^\star) - f(x) = \max_{i=1\dots d} |x_i| \geq \|x^\star - x\|_2 / \sqrt{d}$ for all $x \in \mathcal{X}$, it is easy to see that Condition 2.16 is satisfied with $\kappa = 1$ and $c_\kappa = 1/\sqrt{d}$. Therefore, running LIPO with any $k \geq 1$ would provide an exponential decay of order $O^*_{\mathbb{P}}(e^{-n/2(16kd)^d})$.

We point out that the polynomial bounds of Theorem 2.17 can also be derived from the previous examples by slightly adapting the test functions. For instance, it is easy to see that the function $f(x) = 1 - \|x\|_2^\kappa$ satisfies Condition 2.16 with $\kappa \geq 0$ and $c_\kappa = 1$.

2.3.4 Comparison with existing works

Algorithms. The Piyavskii algorithm (Piyavskii (1972)) is a sequential algorithm which requires the knowledge of the Lipschitz constant $k \geq 0$ and iteratively evaluates the function over a point $X_{t+1} \in \arg\max_{x \in \mathcal{X}} UB_{k,t}(x)$ that maximizes the upper bound on possible values $UB_{k,t}(x) := \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|$ displayed on Figure 2.2. Munos (2014) also proposed a similar algorithm (DOO) that requires the knowledge of a semi-metric $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ for which the function is at least locally smooth (*i.e.*, $\forall x \in \mathcal{X}$, $f(x^\star) - f(x) \leq \ell(x, x^\star)$) and a hierarchical partitioning of the space $\mathcal{X}$ in order to sequentially expand and evaluate the function over the center of a partition which has recorded the highest upper bound computed according to the semi-metric ℓ. With regards to those works, the LIPO algorithm only aims at optimizing globally Lipschitz functions and combines space-filling and exploitation rather than pure exploitation. Recall indeed that LIPO evaluates the function over any point which might improve the function values (see Lemma 2.9) while DOO and the Piyavskii algorithm sequentially select among a restricted set of points the next evaluation point which have recorded the highest upper bound on possible values.

Comparison. To the best of our knowledge, only the consistency of the Piyavskii algorithm was proven in Mladineo (1986) and Munos (2014) derived finite-time upper bounds for DOO with the use of weaker local smoothness assumptions. To cast their results into our framework, we thus considered DOO tuned with the semi-metric $\ell(x,x') = k \cdot \|x - x'\|_2$ over the domain $\mathcal{X} = [0,1]^d$ partitioned into a 2^d-ary tree of hypercubes and with f belonging to the sets of globally smooth functions: (a) $\text{Lip}(k)$, (b) $\mathcal{F}_\kappa =$

$\{f \in \mathrm{Lip}(k)$ satisfying Condition 2.16 with $c_\kappa, \kappa \geq 1\}$ and (c) $\mathcal{F}'_\kappa = \{f \in \mathcal{F}_\kappa : \exists c_2 > 0,\ f(x^\star) - f(x) \leq c_2 \|x - x^\star\|_2^\kappa\}$. The results of the comparison can be found in Table 2.2. Additionally to the novel lower bounds and the rate over $\mathrm{Lip}(k)$, we were able to obtain similar upper bounds as DOO over $\mathcal{F}_\kappa$, uniformly better rates for the functions in $\mathcal{F}'_\kappa$ locally equivalent to $\|x^\star - x\|_2^\kappa$ with $\kappa > 1$ and up to a constant factor a similar exponential rate when $\kappa = 1$. Hence, when f is only known to be k-Lipschitz for some $k \geq 0$, one thus should expect the algorithm exploiting the global smoothness (LIPO) to perform asymptotically better or at least similarly to the one using the local smoothness (DOO) or no information (PRS). However, keeping in mind that the constants provided in the rates are not necessarily optimal, it is interesting to note that the term $(k\sqrt{d}/c_\kappa)^d$ appearing in both the fast rates of LIPO and DOO tends to suggest that if f is also known to be locally smooth for some $k_\ell \ll k$, then the algorithm exploiting the local smoothness k_ℓ is expected to be asymptotically faster than the one using the global smoothness k in the case where $\kappa = 1$.

Algorithm	**DOO**	**LIPO**	**Piyavskii**	**PRS**
$f \in \mathrm{Lip}(k)$				
Consistency	✓	✓	✓	✓
Upper Bound	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$
$f \in \mathcal{F}_\kappa, \kappa > 1$				
Upper bound	$O(n^{-\frac{\kappa}{d(\kappa-1)}})$	$O^*_{\mathbb{P}}(n^{-\frac{\kappa}{d(\kappa-1)}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$
Lower bound	-	$\Omega^*_{\mathbb{P}}(e^{-\frac{\kappa}{d}n})$	-	$\Omega_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$
$f \in \mathcal{F}'_\kappa, \kappa > 1$				
Upper bound	$O(n^{-\frac{\kappa}{d(\kappa-1)}})$	$O^*_{\mathbb{P}}(n^{-\frac{\kappa\times\kappa}{d(\kappa-1)}})$	-	$O_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$
Lower bound	-	$\Omega^*_{\mathbb{P}}(e^{-\frac{\kappa}{d}n})$	-	$\Omega_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$
$f \in \mathcal{F}'_\kappa, \kappa = 1$				
Upper bound	$O(e^{-\frac{n\ln(2)}{(2k\sqrt{d}/c_\kappa)^d}})$	$O^*_{\mathbb{P}}(e^{-\frac{n\ln(2)}{2(16k\sqrt{d}/c_\kappa)^d}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$
Lower bound	-	$\Omega^*_{\mathbb{P}}(e^{-\frac{n}{d}})$	-	$\Omega_{\mathbb{P}}(n^{-\frac{1}{d}})$

Table 2.1: Comparison of the results reported over the difference $\max_{x\in\mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i)$ in global optimization literature. Dash symbols are used when no results could be found.

2.4 Optimization with unknown Lipschitz constant

In this section, we consider the problem of optimizing any unknown function f in the class $\bigcup_{k\geq 0}\mathrm{Lip}(k)$.

2.4.1 The adaptive algorithm

The AdaLIPO algorithm (displayed in Figure 2.4) is an extension of LIPO which involves an estimate of the Lipschitz constant. Instead of the knowledge of a constant $k \geq 0$, it takes

Input: $n \in \mathbb{N}^\star$, $k_{i\in\mathbb{Z}}$, $\mathcal{X} \subset \mathbb{R}^d$, $f \in \bigcup_{k\geq 0} \text{Lip}(k)$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$, $\hat{k}_1 \leftarrow 0$

2. Iterations: Repeat while $t < n$
Let $B_{t+1} \sim \mathcal{B}(p)$
If $B_{t+1} = 1$ (Exploration)
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $B_{t+1} = 0$ (Exploitation)
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X}_{\hat{k}_t,t})$ where $\mathcal{X}_{\hat{k}_t,t}$ denotes the set of potential maximizers of Definition 2.8 computed with k set to $\hat{k}_t$

Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
Let $\hat{k}_t := \inf\left\{ k_{i\in\mathbb{Z}} : \max_{i\neq j} \frac{|f(X_i) - f(X_j)|}{\|X_i - X_j\|_2} \leq k_i \right\}$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Figure 2.4: The AdaLIPO algorithm

as input a parameter $p \in (0,1)$ and a nondecreasing sequence of Lipschitz constant $k_{i\in\mathbb{Z}}$ defining a meshgrid of $\mathbb{R}^+$ (*i.e.* such that $\forall x > 0$, $\exists i \in \mathbb{Z}$ with $k_i \leq x \leq k_{i+1}$). The algorithm is initialized with a Lipschitz constant $\hat{k}_1$ set to 0 and randomly alternates between two distinct phases: exploration and exploitation. Indeed, at each step $t < n$, a Bernoulli random variable B_{t+1} of parameter p which drives the exploration/ exploitation trade-off is sampled. If $B_{t+1} = 1$, the algorithm explores the space by evaluating the function over a point uniformly sampled over $\mathcal{X}$. Otherwise, if $B_{t+1} = 0$, the algorithm exploits the previous evaluations by making an iteration of the LIPO algorithm with the smallest Lipschitz constant of the sequence $\hat{k}_t$ associated with a subset of Lipschitz functions that probably contains f (step abbreviated in the algorithm by $X_{t+1} \sim \mathcal{U}(\mathcal{X}_{\hat{k}_t,t})$). Once an evaluation has been made, the Lipschitz constant estimate $\hat{k}_t$ is updated.

Remark 2.19. (Examples of meshgrids) *Several sequences of Lipschitz constants with various shapes could be considered such as* $k_i = |i|^{sgn(i)}$, $\ln(1+|i|^{sgn(i)})$ *or* $(1+\alpha)^i$ *for any* $\alpha > 0$. *In particular, it should be noticed that the computation of the estimate is straightforward with these sequences. For instance, when* $k_i = (1+\alpha)^i$, *we have* $\hat{k}_t = (1+\alpha)^{i_t}$ *where* $i_t = \lceil \ln(\max_{i\neq j} |f(X_j) - f(X_l)| / \|X_j - X_l\|_2) / \ln(1+\alpha) \rceil$.

Remark 2.20. (Alternative Lipschitz constant estimate) *Due to the generecity of the algorithm, we point out that any Lipschitz constant estimate such as the one proposed by Wood and Zhang (1996) or Bubeck et al. (2011) could also be considered to implement the algorithm. However, as the subsequent analysis requires the estimate to be universally consistent (see Section below), we will only consider the proposed one that presents such a property.*

2.4.2 Convergence analysis

Lipschitz constant estimate. Before starting the analysis of the algorithm, we first provide a control on the Lipschitz constant estimate based on a sample of random evaluations that will be useful to analyze its performance. Precisely, the next result illustrates the purpose of using a discretization of Lipschitz constants instead of a raw estimate of the maximum slope by showing that, given this estimate, a small subset of functions containing the unknown function can be recovered in a finite-time.

Proposition 2.21. *Let $f \in \bigcup_{k\geq 0}\mathrm{Lip}(k)$ be any non-constant Lipschitz function. Then, if $\hat{k}_t$ denotes the Lipschitz constant estimate of Algorithm 2 computed with any increasing sequence $k_{i\in\mathbb{Z}}$ defining a meshgrid of $\mathbb{R}^+$ over a sample $(X_1, f(X_1)), \dots, (X_t, f(X_t))$ of $t \geq 2$ evaluations where $X_1, \dots, X_t$ are uniformly and independently distributed over $\mathcal{X}$, we have that*

$$\mathbb{P}\left(f \in \mathrm{Lip}(\hat{k}_t)\right) \geq 1 - (1 - \Gamma(f, k_{i^\star-1}))^{\lfloor t/2 \rfloor}$$

where the coefficient

$$\Gamma(f, k_{i^\star-1}) := \mathbb{P}\left(\frac{|f(X_1) - f(X_2)|}{\|X_1 - X_2\|_2} > k_{i^\star-1}\right) > 0$$

with $i^\star = \min\{i \in \mathbb{Z} : f \in \mathrm{Lip}(k_i)\}$, is strictly positive.

The following remarks provide some insights on the quantities involved in the bound.

Remark 2.22. (Measure of global smoothness) *The coefficient $\Gamma(f, k_{i^\star-1})$ which appears in the lower bound of Proposition 2.21 can be seen as a measure of the global smoothness of the function f with regards to $k_{i^\star-1}$. Indeed, observing that $(1/\lfloor t/2 \rfloor) \cdot \sum_{i=1}^{\lfloor t/2 \rfloor} \mathbb{I}\{|f(X_i) - f(X_{i+\lfloor t/2 \rfloor})| > k_{i^\star-1}\|X_i - X_{\lfloor t/2 \rfloor + i}\|_2\} \xrightarrow{p} \Gamma(f, k_{i^\star-1})$, it is easy to see that this coefficient records the ratio of volume the product space $\mathcal{X} \times \mathcal{X}$ where f is witnessed to be at least $k_{i^\star-1}$-Lipschitz.*

Remark 2.23. (Density of the sequence of Lipschitz constants) *As a consequence of the previous remark, we point out that the density of the sequence of Lipschitz constants $k_{i\in\mathbb{Z}}$, captured here by $\alpha = \sup_{i\in\mathbb{Z}}(k_{i+1} - k_i)/k_i$, has opposite impacts on the maximal deviation and the convergence rate of the estimate. Indeed, as α is involved in both the following upper bounds on the deviation and on the coefficient Gamma: $(\lim_{t\to\infty} \hat{k}_t - k^\star)/k^\star \leq \alpha$ and $\Gamma(f, k_{i^\star-1}) \leq \Gamma(f, k^\star/(1+\alpha))$ where $k^\star = \sup\{k \geq 0 : f \notin \mathrm{Lip}(k)\}$ denotes the optimal Lipschitz constant, we deduce that using a dense sequence of Lipschitz constants with a small α tends to reduce the bias but also the convergence rate through a small coefficient $\Gamma(f, k_{i^\star-1})$.*

Remark 2.24. (Impact of the dimensionality) *Last, we provide a simple result which illustrates the fact that, independently of the Lipschitz constant, the task of estimating the constant becomes harder as the dimensionality d grows large. Let $f : [0,1]^d \to \mathbb{R}$ be any k-Lipschitz function with regards to the Euclidean distance for some $k \geq 0$ and let X and X' be two random variables uniformly distributed over $[0,1]^d$. Then, using the dimension free concentration property used in the introduction of Bobkov (2010), we directly obtain that for all $\lambda \in (0,1)$,*

$$\mathbb{P}\left\{\frac{|f(X) - f(X')|}{\|X - X'\|_2} > \lambda k\right\} \leq 5e^{-cd\lambda^2}$$

where $c > 0$ is an absolute constant. Combining this result with the naive bound $\mathbb{P}(f \in \mathrm{Lip}(\hat{k}_t)) \leq t^2\Gamma(f, k_{i^\star-1})$ shows that $\Omega_{\mathbb{P}}(e^{cdk/k_{i^\star-1}})$ evaluation points should at least be collected in order to successfully estimate the Lipschitz constant.

Equipped with Proposition 2.21, we may now analyze the convergence properties of AdaLIPO.

Analysis of AdaLIPO. Given the consistency equivalence of Proposition 2.3, one can directly obtain the following asymptotic result.

Proposition 2.25. (Consistency) *The AdaLIPO algorithm tuned with any parameter $p \in (0,1)$ and any sequence of Lipschitz constant $k_{i\in\mathbb{Z}}$ defining a meshgrid of $\mathbb{R}^+$ is consistent over the set of Lipschitz functions, i.e.,*

$$\forall f \in \bigcup_{k\geq 0}\mathrm{Lip}(k),\ \max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x).$$

The next result provides a first finite-time bound on the difference between the maximum and its approximation.

Proposition 2.26. (Upper Bound) *Consider AdaLIPO tuned with any $p \in (0,1)$ and any sequence $k_{i\in\mathbb{Z}}$ defining a meshgrid of $\mathbb{R}^+$. Then, for any non-constant $f \in \bigcup_{k\geq 0}\mathrm{Lip}(k)$, any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k_{i^\star} \times \mathrm{diam}(\mathcal{X}) \times \left(\frac{5}{p} + \frac{2\ln(\delta/3)}{p\ln(1-\Gamma(f,k_{i^\star\text{-}1}))}\right)^{\frac{1}{d}} \times \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}}$$

where $\Gamma(f, k_{i^\star-1})$ and $i^\star$ are defined as in Proposition 2.21 and $\ln(0) = -\infty$ by convention.

This result might be misleading since it advocates that doing pure exploration gives the best rate (*i.e.*, when $p \to 1$). As Proposition 2.21 provides us with the guarantee that $f \in \mathrm{Lip}(\hat{k}_t)$ within a finite number of iterations, one can however recover faster convergence rates similar to the one reported for LIPO where the Lipschitz constant is assumed to be known.

Theorem 2.27. (Fast Rates) *Consider the same assumptions as in Proposition 2.26 and assume in addition that f satisfies Condition 2.16 for some $\kappa \geq 1$, $c_\kappa \geq 0$. Then, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k_{i^\star} \times \mathrm{diam}(\mathcal{X}) \times \exp\left(\frac{2\ln(\delta/4)}{p\ln(1-\Gamma(f,k_{i^\star-1}))} + \frac{7\ln(4/\delta)}{p(1-p)^2}\right) \times \begin{cases} \exp\left\{-C_{k_{i^\star},\kappa} \cdot \dfrac{n\,(1-p)\ln(2)}{2\ln(n/\delta) + 4(2\sqrt{d})^d}\right\}, & \kappa = 1 \\ 2^\kappa \left(1 + C_{k_{i^\star},\kappa} \cdot \dfrac{n(1-p)(2^{d(\kappa-1)}-1)}{2\ln(n/\delta) + 4(2\sqrt{d})^d}\right)^{-\frac{\kappa}{d(\kappa-1)}}, & \kappa > 1 \end{cases}$$

where $C_{k_{i^\star}\kappa} = (c_\kappa \max_{x\in\mathcal{X}} \|x - x^\star\|_2^{\kappa-1} / 8k_{i^\star})^d$.

This bound shows the precise impact of the parameters p and $k_{i\in\mathbb{Z}}$ on the convergence of the algorithm. It illustrates the complexity of the exploration/exploitation trade-off through a constant term and a convergence rate which are inversely correlated to the exploration parameter and the density of the sequence of Lipschitz constants. Last, it should also be noticed that the examples of Section 2.3.3 still remain valid as the bounds derived here are of the same order as when k is known (LIPO). We may compare the algorithm to existing works.

2.4.3 Comparison with previous works

Algorithms. The DIRECT algorithm (Jones et al. (1993)) is a Lipschitz optimization algorithm where the Lipschitz constant is unknown. It uses a deterministic splitting technique of the search space in order to sequentially divide and evaluate the function over a subdivision of the space that have recorded the highest upper bound among all subdivisions of similar size for at least a possible value of k. Munos (2014) generalized DIRECT in a broader setting by extending the the DOO algorithm to any unknown and arbitrary local semi-metric, under the name SOO. With regards to these works, we proposed an alternative stochastic strategy which directly relies on the estimation of the Lipschitz constant and thus only presents guarantees for globally Lipschitz functions.

Comparison. Up to our knowledge, only the consistency property of DIRECT was shown in Finkel and Kelley (2004) and Munos (2014) derived convergence rates for SOO using

Algorithm	**AdaLIPO**	**DIRECT**	**PRS**	**SOO**
$f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$				
Consistency	✓	✓	✓	✓
Upper Bound	$O^*_{\mathbb{P}}(n^{-\frac{1}{d}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$	-
$f \in \mathcal{F}_\kappa$, $\kappa > 1$				
Upper bound	$O^*_{\mathbb{P}}(n^{-\frac{\kappa}{d(\kappa-1)}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$	$O(n^{-\frac{\kappa}{2d(\kappa-1)}})$
Lower bound	-	-	$\Omega_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$	-
$f \in \mathcal{F}'_\kappa$, $\kappa > 1$				
Upper bound	$O^*_{\mathbb{P}}(n^{-\frac{\kappa\times\kappa}{d(\kappa-1)}})$	-	$O_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$	$O(e^{-\frac{\sqrt{n}\ln(2)}{(\sqrt{2d}(c_2/c_\kappa)^{1/\kappa})^d}})$
Lower bound	-	-	$\Omega_{\mathbb{P}}(n^{-\frac{\kappa}{d}})$	-
$f \in \mathcal{F}'_\kappa$, $\kappa = 1$				
Upper bound	$O^*_{\mathbb{P}}(e^{-\frac{n(1-p)\ln(2)}{4(16k_{i^\star}\sqrt{d}/c_\kappa)^d}})$	-	$O_{\mathbb{P}}(n^{-\frac{1}{d}})$	$O(e^{-\frac{\sqrt{n}\ln(2)}{(c_2\sqrt{2d}/c_\kappa)^d}})$
Lower bound	-	-	$\Omega_{\mathbb{P}}(n^{-\frac{1}{d}})$	-

Table 2.2: Comparison of the results reported over the difference $\max_{x\in\mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i)$ in global optimization literature. Dash symbols are used when no results could be found.

weaker local smoothness assumptions. To compare our results, we considered SOO tuned with the depth function $h_{\max}(t) = \sqrt{t}$ providing the best rate, over the domain $\mathcal{X} = [0,1]^d$ partitioned into a 2^d-ary tree of hypercubes and with f belonging to the following sets of globally Lipschitz functions: (a) $\bigcup_{k\geq 0} \text{Lip}(k)$, (b) $\mathcal{F}_\kappa = \{f \in \bigcup_{k\geq 0} \text{Lip}(k)$ satisfying Condition 2.16 with $c_\kappa, \kappa \geq 1\}$ and (c) $\mathcal{F}'_\kappa = \{f \in \mathcal{F}_\kappa : f(x^\star) - f(x) \leq c_2 \|x - x^\star\|_2^\kappa\}$. The result of the comparison can be found in Table 2.2. Additionally to the novel rate over the class of Lipschitz functions, we were also able to obtain a faster polynomial rate than SOO over the set $\mathcal{F}_\kappa$. However, SOO achieves its best rate of order $O(e^{-c\sqrt{n}})$ over the whole set $\{\mathcal{F}'_\kappa, \kappa \geq 1\}$ by adapting to any function locally equivalent to $\|x^\star - x\|_2^\kappa$ while AdaLIPO achieves a slower polynomial rate when $\kappa > 1$, but an even faster exponential rate of order $O^\star_{\mathbb{P}}(e^{-cn})$ when $\kappa = 1$. Thus, by exploiting the global smoothness of the function (AdaLIPO), we were able to derive a faster convergence rate than the best one reported for the algorithm exploiting the local smoothness (SOO) which however remains valid over a larger set of functions.

2.5 Discussion and perspectives

In this chapter, we introduced two novel strategies for global optimization based on a sequential exploitation of the global smoothness of the unknown function: LIPO which requires the knowledge of the Lipschitz constant and its adaptive version AdaLIPO which estimates the Lipschitz constant along with the optimization. A theoretical analysis is provided in this Chapter and empirical results based on synthetic and real problems are provided in Chapter 4, demonstrating the competitiveness of the adaptive algorithm with regards to existing state-of-the-art global optimization methods. Possible future research directions include:

- Deriving problem-dependent lower bounds and refining the bounds of Theorem 2.18 in order to match the bounds of Theorem 2.17.

- Investigating alternative covering methods and determining better values for the exploration/exploitation parameter p. For instance, an approach would consist in using dynamic values for p_t that depend on the previous evaluations and using near-random covering methods such as the Sobol sequences (Sobol (1967)) or the Latin Hypercube Sampling (McKay et al. (1979)).

- Extending the algorithms to the noisy scenario and investigating alternative sampling strategies. For instance, a plausible strategy with fixed $k \geq 0$ would consist in sampling the next evaluation point $X_{t+1} \sim \mathcal{U}(\{x \in \mathcal{X} : UB_{k,t}(x) \geq \max_{x\in\mathcal{X}} UB_{k,t}(x) - u \times (\max_{x\in\mathcal{X}} UB_{k,t}(x) - \max_{i=1\dots t} f(X_i))\})$ among the set of points with an upper bound high enough for some $u \in [0,1]$. In this case, the parameter u would control the randomness of the strategy and it could lead to more or less selective strategies. Note that, in this case, the LIPO strategy is obtained when $u = 1$ while the deterministic Piyavskii (1972) algorithm is obtained when $u = 0$.

- A finer investigation on the price of using a global smoothness assumption (LIPO) instead of a local one (DOO). Although a simple comparison of the convergence rates obtained for these algorithms can be found in Section 2.3.4, we point out that an analysis beyond the big O would require a precise comparison of the constants

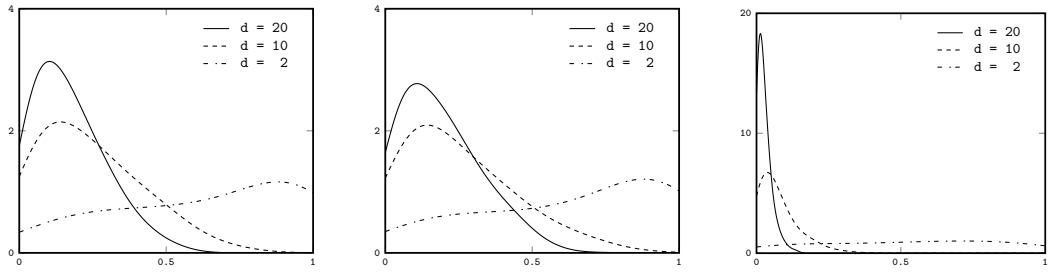


Figure 2.5: Density function of the random slope $|f(X) - f(X')| / \|X - X'\|_2$ where $X, X' \overset{\text{iid}}{\sim} \mathcal{U}([0,1]^d)$ for various dimensionality d on (a) the sphere function $f(x) = 1 - \|x\|_2$, (b) the linear slope $f(x) = 1 - \sum_{i=1}^{d} x_i/\sqrt{d}$, (c) the largest coordinate function $f(x) = 1 - \max\{x_1, \ldots, x_d\}$.

obtained for each algorithm that are not necessarily optimal and available under the same assumptions at this point.

- A last open question would be to determine whether exploiting the global smoothness of the function can lead to efficient algorithm when the dimensionality of the input space is large. Recall indeed that (i) although the fast rates of order $O_{\mathbb{P}}(e^{-cn/(\sqrt{d})^d})$ we derive in Section 2.3.3 are exponential, they strongly depend on the dimensionality and (ii) the bound reported in Remark 2.24 shows that, independently of the constant k, the random slope of any Lipschitz function concentrates towards zero with the dimension: *i.e.*, for any k-Lipschitz function $f : [0,1]^d \to \mathbb{R}$, we have for all $\lambda \in (0,1)$,

$$\mathbb{P}\left\{ \frac{|f(X) - f(X')|}{\|X - X'\|_2} > \lambda k \right\} \leq 5e^{-cd\lambda^2}$$

where X and X' are uniformly distributed over $[0,1]^d$. These simple statements suggest that using the Lipschitz constant of a function tends to be less efficient and informative as the dimensionality grows large. To illustrate this phenomenon, the density functions of the random slope of three 1-Lipschitz functions with various dimensionalities are displayed in Figure 2.5. It would thus be interesting to investigate for which range of dimensionalities the methods proposed in the chapter can display competitive results with regards to other optimization methods.

2.6 Proofs

In this section, we provide the proofs of the results presented in this chapter.

2.6.1 Proofs of Section 2.2

We provide here the proofs of Propositions 2.3, 2.5, 2.6 and Example 2.4.

Proof of proposition 2.3. ($\Leftarrow$) Let A be any global optimization algorithm such that $\forall f \in \bigcup_{k\geq 0} \text{Lip}(k)$, $\sup_{x\in\mathcal{X}} \min_{i=1\ldots n} \|X_i - x\|_2 \xrightarrow{p} 0$. Pick any $\epsilon > 0$, pick any $f \in \bigcup_{k\geq 0} \text{Lip}(k)$

and let $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$ be the corresponding level set. As $\mathcal{X}_\epsilon$ is non-empty since f is Lipschitz continuous, there necessarily exists some $x_\epsilon \in \mathcal{X}$ and $r_\epsilon > 0$ such that $B(x_\epsilon, r_\epsilon) \cap \mathcal{X} \subseteq \mathcal{X}_\epsilon$. Therefore, if $X_1, \dots, X_n$ denotes a sequence a sequence of n evaluation points generated by A over f, using the convergence in probability of the mesh grid directly gives that

$$\begin{aligned}
\mathbb{P}\left(\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) > \epsilon\right) &= \mathbb{P}\left(\bigcap_{i=1}^{n} \{X_i \notin \mathcal{X}_\epsilon\}\right) \\
&\leq \mathbb{P}\left(\bigcap_{i=1}^{n} \{X_i \notin B(x_\epsilon, r_\epsilon) \cap \mathcal{X}\}\right) \\
&= \mathbb{P}\left(\min_{i=1\dots n} \|X_i - x_\epsilon\|_2 > r_\epsilon\right) \\
&\leq \mathbb{P}\left(\sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|X_i - x\|_2 > r_\epsilon\right) \xrightarrow[n\to\infty]{} 0.
\end{aligned}$$

($\Rightarrow$) Let A be any global optimization algorithm consistent over the set of Lipschitz functions and assume by contradiction that there exists some function $f^\star \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$ such that $\sup_{x\in_X} \min_{i=1\dots n} \|x - X_i\|_2 \overset{p}{\not\to} 0$. We prove the implication in two steps: first, we show that there exists a ball $B(c^\star, \epsilon)$ for some $c^\star \in \mathcal{X}$ which is almost never hit by the algorithm and second, we build a Lipschitz function which admits its maximum over this ball.

First step. Let $\{X_i\}_{i\in\mathbb{N}^\star}$ be a sequence of evaluation points generated by A over $f^\star$. Observe first that since for all $\epsilon > 0$, the series $n \in \mathbb{N}^\star \mapsto \mathbb{P}(\sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|x - X_i\|_2 > \epsilon)$ is non-increasing, then it necessarily follows from the contradiction assumption that

$$\exists \epsilon_1, \epsilon_2 > 0 \text{ such that } \forall n \in \mathbb{N}^\star,\ \mathbb{P}\left(\sup_{x\in\mathcal{X}} \min_{i=1\dots n} \|x - X_i\|_2 > \epsilon_1\right) > \epsilon_2. \tag{2.1}$$

We may now show that a ball is almost nevet hit by the algorithm with probability at least $\epsilon_2/(2\mathcal{N}_{\epsilon_1}(\mathcal{X}))$. Consider any sequence $c_1, \dots, c_{N_1}$ of $N_1 = \mathcal{N}_{\epsilon_1}(\mathcal{X})$ points in $\mathcal{X}$ defining an ϵ_1-cover of $\mathcal{X}$ and suppose by contradiction that

$$\forall c \in \{c_1, \dots, c_{N_1}\},\ \exists n_c \in \mathbb{N}^\star \text{ such that } \mathbb{P}\left(\bigcap_{i=1}^{n_c} \{X_i \notin B(c, \epsilon_1) \cap \mathcal{X}\}\right) \leq \frac{\epsilon_2}{2N_1}.$$

Setting $N_2 = \max_{c\in\{c_1,\cdots,c_{N_1}\}} n_c$, we have that

$$\forall c \in \{c_1, \dots, c_{N_1}\},\ \mathbb{P}\left(\bigcap_{i=1}^{N_2} \{X_i \notin B(c, \epsilon) \cap \mathcal{X}\}\right) \leq \frac{\epsilon_2}{2N_1}.$$

However, as $c_1,\dots,c_{N_1}$ form an ϵ_1-cover of $\mathcal{X}$, it necessarily follows that

$$\begin{aligned}
\mathbb{P}\left(\sup_{x\in\mathcal{X}}\min_{i=1\dots N_2}\|x-X_i\|_2\leq\epsilon_1\right) &\geq \mathbb{P}\left(\bigcap_{j=1}^{N_1}\bigcup_{i=1}^{N_2}\{X_i\in B(c_j,\epsilon_1)\cap\mathcal{X}\}\right)\\
&= 1-\mathbb{P}\left(\bigcup_{j=1}^{N_1}\bigcap_{i=1}^{N_2}\{X_i\notin B(c_j,\epsilon_1)\cap\mathcal{X}\}\right)\\
&\geq 1-\sum_{j=1}^{N_1}\mathbb{P}\left(\bigcap_{i=1}^{N_2}\{X_i\notin B(c_j,\epsilon)\cap\mathcal{X}\}\right)\\
&\geq 1-N_1\times\frac{\epsilon_2}{2N_1}\\
&= 1-\frac{\epsilon_2}{2}
\end{aligned}$$

which contradicts (2.1). Therefore, we deduce that

$$\exists c^\star\in\{c_1,\dots,c_{N_\epsilon}\}\text{ such that }\forall n\in\mathbb{N}^\star,\ \mathbb{P}\left(\bigcap_{i=1}^{n}\{X_i\notin B(c^\star,\epsilon_1)\cap\mathcal{X}\}\right)\geq\frac{\epsilon_2}{2N_1}.$$

Second Step. Using this center $c^\star\in\mathcal{X}$, one can then introduce the function $\tilde{f}:\mathcal{X}\mapsto\mathbb{R}$ defined for all $x\in\mathcal{X}$ by

$$\tilde{f}(x)=\begin{cases}f^\star(x)+3\left(1-\frac{\|c^\star-x\|_2}{\epsilon_1}\right)\times\left(\max_{x\in\mathcal{X}}f^\star(x)-\min_{x\in\mathcal{X}}f^\star(x)\right) & \text{if } x\in B(c^\star,\epsilon_1)\\ f^\star(x) & \text{otherwise}\end{cases}$$

which is maximized over $B(c^\star,\epsilon_1)$ and Lipschitz continuous since both $f^\star$ and $x\mapsto\|c^\star-x\|_2$ are Lipschitz. However, as $\tilde{f}$ and $f^\star$ can not be distinguished over $\mathcal{X}/B(c,\epsilon_1)$, we have that $\forall n\in\mathbb{N}^\star$,

$$\begin{aligned}
\mathbb{P}\left(\max_{x\in\mathcal{X}}\tilde{f}(x)-\max_{i=1\dots n}\tilde{f}(X_i')>\max_{x\in\mathcal{X}}f(x)\right) &\geq \mathbb{P}\left(\bigcap_{i=1}^{n}\{X_i'\notin B(c,\epsilon_1)\cap\mathcal{X}\}\right)\\
&= \mathbb{P}\left(\bigcap_{i=1}^{n}\{X_i\notin B(c,\epsilon_1)\cap\mathcal{X}\}\right)\\
&\geq \epsilon_2/(2N_1)\\
&> 0
\end{aligned}$$

where $X_1',\dots,X_n'$ denotes a sequence of evaluation points generated by A over $\tilde{f}$. Hence, we deduce that that there exists some function $\tilde{f}\in\bigcup_{k\geq0}\text{Lip}(k)$ such that $\max_{i=1\dots n}\tilde{f}(X_i')\overset{p}{\nrightarrow}\max_{x\in\mathcal{X}}\tilde{f}(x)$, which contradicts the fact that A is consistent over $\bigcup_{k\geq0}\text{Lip}(k)$. We thus deduce that if A is consistent over the set of Lipschitz functions, we necessarily have that $\sup_{x\in\mathcal{X}}\min_{i=1\dots n}\|X_i-x\|\overset{p}{\to}0$ for all $f\in\bigcup_{k\geq0}\text{Lip}(k)$. □

Proof of Example 2.4. This proof is postponed to the Appendix Section where the Pure

Random Search is analyzed. □

Proof of Proposition 2.5. The proof heavily builds upon the arguments used in the proof of Theorem 1 in Bull (2011). Pick any algorithm $A \in \mathcal{A}$ and any constant $C > 0$. Fix any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$ and set $N_\delta = \lceil (n/\delta)^{1/d} \rceil$. By definition of $\mathrm{rad}(\mathcal{X})$, we know there exists some $x \in \mathcal{X}$ such that $x + [0, 2\mathrm{rad}(\mathcal{X})/\sqrt{d}]^d \subseteq \mathcal{X}$. One can then define for all $I \in \{1, \dots, N_\delta\}^d$, the centers c_I of the hypercubes H_I whose sides are equal to $D = 2\mathrm{rad}(\mathcal{X})/(\sqrt{d} N_\delta)$ and cover $\mathcal{X}$ (*i.e.*, $\bigcup_I H_I = x + [0, 2\mathrm{rad}(\mathcal{X})/\sqrt{d}]^d \subseteq \mathcal{X}$). Now, let $X_1, \dots, X_n$ be a sequence of n evaluation points generated by the algorithm A over the constant zero function $f_0 : x \in \mathcal{X} \mapsto 0$ and define for all $I \in \{1, \dots, N_\delta\}^d$ the event

$$E_I = \bigcap_{i=1}^{n} \{X_i \notin \mathrm{Int}(H_I)\}$$

where $\mathrm{Int}(\mathcal{X})$ denotes the interior of $\mathcal{X}$. As the interiors of the N_δ^d hypercubes are disjoint and we have n points, it necessarily follows that

$$N_\delta^d \times \max_I \mathbb{P}(E_I) \geq \sum_I \mathbb{P}(E_I) = \mathbb{E}\left[\sum_I \mathbb{I}\{E_I\}\right] \geq N_\delta^d - n.$$

Hence, there exists some fixed $I^\star$ only depending on $\mathcal{A}$ which maximizes the above probability, and thus satisfies

$$\mathbb{P}(E_{I^\star}) \geq \frac{N_\delta^d - n}{N_\delta^d} = 1 - \frac{n}{\lceil (n/\delta)^{1/d} \rceil^d} \geq 1 - \delta.$$

Now, using the center $c_{I^\star}$ of the hypercube $H_{I^\star}$, one can introduce the function $\tilde{f} \in \bigcup_{k \geq 0} \mathrm{Lip}(k)$ defined for all $x \in \mathcal{X}$ by

$$\tilde{f}(x) = \begin{cases} C \times (1 - 2\|c_{I^\star} - x\|_2 / D) & \text{if } \|c_{I^\star} - x\|_2 \leq D/2 \\ 0 & \text{otherwise,} \end{cases}$$

which is maximized an equal to C over $c_{I^\star}$. However, since the functions $\tilde{f}$ and f_0 can not be distinguished over $\mathcal{X} / H_{I^\star}$, we have that

$$\mathbb{P}\left(\max_{x \in X} \tilde{f}(x) - \max_{i=1\dots n} \tilde{f}(X_i') \geq C\right) \geq \mathbb{P}\left(\bigcap_{i=1}^{n} \{X_i' \notin \mathrm{Int}(H_{I^\star})\}\right) = \mathbb{P}(E_{I^\star}) \geq 1 - \delta$$

where $X_1', \dots, X_n'$ denotes a sequence of evaluation points generated by $\mathcal{A}$ over $\tilde{f}$ and proves the result. □

Proof of Proposition 2.6. (Lower bound). Pick any $n \in \mathbb{N}^\star$ and set $D = 2\mathrm{rad}(\mathcal{X})/(\sqrt{d}\lceil (2n)^{1/d} \rceil)$. It can easily be shown by reproducing the same steps as in the proof of Proposition 2.5 with δ set to $1/2$, that for any global optimization algorithm A, there exists a function $\tilde{f}_A \in \mathrm{Lip}(k)$ defined by

$$\tilde{f}_A(x) = \begin{cases} kD/2 - k \cdot \|c_A - x\|_2 & \text{if } \|c_A - x\|_2 \leq D/2 \\ 0 & \text{otherwise,} \end{cases}$$

for some center $c_A \in \mathcal{X}$ only depending on A, for which we have $\mathbb{P}(\max_{x\in X} \tilde{f}_A(x) - \max_{i=1\dots n} \tilde{f}_A(X_i) \geq k \cdot D/2) \geq 1/2$ where $X_1, \dots, X_n$ is a sequence of n evaluation points generated by A over $\tilde{f}_A$. Therefore, using the definition of the supremum and Markov's inequality gives that $\forall A \in \mathcal{A}$:

$$\begin{aligned}\sup_{f\in \mathrm{Lip}(k)} \mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)\right] &\geq \mathbb{E}\left[\max_{x\in\mathcal{X}} \tilde{f}_A(x) - \max_{i=1\dots n} \tilde{f}_A(X_i)\right] \\ &\geq \frac{kD}{2} \times \mathbb{P}\left(\max_{x\in X} \tilde{f}_A(x) - \max_{i=1\dots n} \tilde{f}_A(X_i) \geq k \cdot \frac{D}{2}\right) \\ &\geq k \cdot \frac{\mathrm{rad}(\mathcal{X})}{8\sqrt{d}} \cdot n^{-\frac{1}{d}}.\end{aligned}$$

As the previous inequaliy holds true for any algorithm A, the proof is complete.

(Upper bound). Sequentially using the fact that (i) the infimum minimax loss taken over all the algorithms is necessarily upper bounded by the loss suffered by a Pure Random Search, (ii) for any positive random variable X, $\mathbb{E}[X] = \int_{t=0}^{\infty} \mathbb{P}(X \geq t)\mathrm{d}t$, (iii) Proposition B.8 provided in the Appendix and (iv) the change of variable $u = n(t/\operatorname{diam}(\mathcal{X}))^{1/d}$, we obtain that

$$\begin{aligned}\inf_{A\in\mathcal{A}} \sup_{f\in \mathrm{Lip}(k)} \mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)\right] &\leq \sup_{f\in \mathrm{Lip}(k)} \mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i')\right] \\ &\leq \int_0^{\infty} \exp\left\{-n(t/k\cdot\operatorname{diam}(\mathcal{X}))^{1/d}\right\}\mathrm{d}t \\ &= k\cdot\operatorname{diam}(\mathcal{X})\cdot n^{-d}\cdot d\cdot\int_0^{\infty} u^{d-1}e^{-u}\mathrm{d}u \\ &= k\cdot\operatorname{diam}(\mathcal{X})\cdot n^{-d}\cdot d\cdot\Gamma(d)\end{aligned}$$

where $X_1', \dots, X_n'$ is a sequence of n independent copies of $X' \sim \mathcal{U}(\mathcal{X})$ and $\Gamma(\cdot)$ denotes the Euler's Gamma function. Recalling that $\Gamma(d) = (d-1)!$ for all $d \in \mathbb{N}^\star$ completes the proof. □

2.6.2 Proofs of Section 2.3

In this section, we provide the proofs for Lemma 2.9, Proposition 2.12, Proposition 2.13, Corollary 2.14, Proposition 2.15, Theorem 2.17 and Theorem 2.18.

Proof of Lemma 2.9. The first implication ($\Rightarrow$) is a direct consequence of the definition of $\mathcal{X}_{k,t}$. Noticing that the function $\hat{f} : x \mapsto \min(\max_{i=1\dots t} f(X_i), \min_{i=1\dots t} f(X_i) + k\|x - X_i\|_2)$ belongs to $\mathcal{F}_{k,t}$ and that $\arg\max_{x\in\mathcal{X}} \hat{f}(x) = \{x \in \mathcal{X} : \min_{i=1\dots t} f(X_i) + k\|x - X_i\|_2 \geq \max_{i=1\dots t} f(X_i)\}$ proves the second implication. □

Proof of Proposition 2.12. Using the result of Proposition 2.13, the proof is straightforward. Fix any $f \in \mathrm{Lip}(k)$, pick any $n \in \mathbb{N}^\star$, set $\epsilon > 0$ and let $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$ be the corresponding level set. Denoting by $X_1', \dots, X_n'$ a sequence of n random variable uniformly distributed over $\mathcal{X}$ and observing that $\mu(\mathcal{X}_\epsilon) > 0$, we directly

obtain from Proposition 2.13 that

$$\begin{aligned}\mathbb{P}\left(\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) > \epsilon\right) &\leq \mathbb{P}\left(\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i') > \epsilon\right) \\ &= \mathbb{P}\left(\bigcap_{i=1}^{n}\{X_i' \notin \mathcal{X}_\epsilon\}\right) \\ &\leq \left(1 - \frac{\mu(\mathcal{X}_\epsilon)}{\mu(\mathcal{X})}\right)^n \xrightarrow[n\to\infty]{} 0.\end{aligned}$$

□

Proof of Proposition 2.13. The proof is similar to the one of Proposition 12 in Malherbe et al. (2016). However, we point out that a detailed proof of this Proposition can be found in the next Chapter (see the proof of Proposition 3.14). □

Proof of Corollary 2.14. This is the standard upper bound achieved by the Pure Random Search. It is obtained by combining Proposition 2.13 and Proposition B.8 stated in the Appendix. □

Proof of Proposition 2.15. Fix any $\delta \in (0,1)$, set $n \in \mathbb{N}^\star$ and $k > 0$ and let $r_{\delta,n} = \mathrm{rad}(\mathcal{X})\,(\delta/n)^{\frac{1}{d}}$ be the value of the lower bound divided by k. As $\mathrm{rad}(\mathcal{X}) > 0$, there necessarily exists some point $x^\star \in \mathcal{X}$ such that $B(x^\star, \mathrm{rad}(\mathcal{X})) \subseteq \mathcal{X}$. Based on this point, one can introduce the function $\tilde{f} \in \mathrm{Lip}(k)$ defined for all $x \in \mathcal{X}$ by

$$\tilde{f}(x) = \begin{cases} k \cdot r_{\delta,n} - k \cdot \|x - x^\star\|_2 & \text{if } x \in B(x^\star, r_{\delta,n}) \\ 0 & \text{otherwise.} \end{cases}$$

Denoting now by $X_1, \dots, X_n$ a sequence of n evaluation points generated by LIPO tuned with a parameter k over $\tilde{f}$ and observing that (i) X_1 is uniformly distributed over $\mathcal{X}$, (ii) X_{i+1} is also uniformly distributed over $\mathcal{X}$ for $i \geq 1$ as soon as only constant evaluations have been recorded (*i.e.* $\mathcal{X}_{k,i+1} = \mathcal{X}$ on the event $\bigcap_{t\leq i}\{X_t \notin B(x^\star, r_{\delta,n})\}$) and (iii) the

definition of $\mathrm{rad}(\mathcal{X})$, we have that

$$\begin{aligned}
\mathbb{P}\left(\tilde{f}(x^\star) - \max_{i=1\dots n} \tilde{f}(X_i) \geq k \cdot r_{\delta,n}\right) &\geq \mathbb{P}\left(\bigcap_{i=1}^{n} \{X_i \notin B(x^\star, r_{\delta,n})\}\right) \\
&= \Bigg[\mathbb{P}(X_1 \notin B(x^\star, r_{\delta,n})) \times \\
&\prod_{i=1}^{n-1} \mathbb{P}\left(X_{i+1} \notin B(x^\star, r_{\delta,n}) \mid \bigcap_{t=1}^{i} \{X_t \notin B(x^\star, r_{\delta,n})\}\right)\Bigg] \\
&= \left(1 - \frac{\mu(B(x^\star, r_{\delta,n}) \cap \mathcal{X})}{\mu(\mathcal{X})}\right)^n \\
&\geq \left(1 - \left(\frac{r_{\delta,n}}{\mathrm{rad}(\mathcal{X})}\right)^d\right)^n \\
&= \left(1 - \frac{\delta}{n}\right)^n \\
&\geq 1 - \delta.
\end{aligned}$$

□

Proof of Theorem 2.17. Pick any $n \in \mathbb{N}^\star$, fix any $\delta \in (0,1)$ and let $X_1, \dots, X_n$ be a sequence of n evaluation points generated by the LIPO algorithm over f after n iterations. To clarify the proof, we set some specific notations: let $D = \max_{x \in \mathcal{X}} \|x - x^\star\|_2$, set

$$M = \begin{cases} \left\lfloor \left(\frac{c_\kappa}{8k}\right)^d \cdot \frac{n}{\ln(n/\delta) + 2(2\sqrt{d})^d} \right\rfloor & \text{if } \kappa = 1 \\ \left\lfloor \frac{1}{\ln(2) d(\kappa - 1)} \ln\left(1 + \left(\frac{c_\kappa D^\kappa}{8kD}\right)^d \frac{n(2^{d(\kappa-1)} - 1)}{\ln(n/\delta) + 2(2\sqrt{d})^d}\right) \right\rfloor & \text{otherwise,} \end{cases}$$

assume without loss of generality that $M \geq 1$ (otherwise the result would trivially hold), define for all $m \in \{1 \dots M\}$ the series of integers:

$$N_m := \left\lceil \sqrt{d} \cdot \left(\frac{8kD}{c_\kappa D^\kappa}\right) \cdot 2^{m(\kappa-1)} \right\rceil^d \quad \text{and} \quad N'_m := \left\lceil \ln(M/\delta) \cdot \left(\frac{8kD}{c_\kappa D^\kappa}\right)^d \cdot 2^{md(\kappa-1)} \right\rceil$$

and let $\tau_0, \dots, \tau_M$ be the series of stopping times initialized by $\tau_0 = 0$ and defined for all $m \geq 1$ by

$$\tau_m := \inf\left\{ t \geq \tau_{m-1} \mid \sum_{i=\tau_{m-1}+1}^{t} \mathbb{I}\left\{X_i \in B(x^\star, 2 \cdot D \cdot 2^{-m})\right\} = N'_m \right\}.$$

The stopping time τ_m correspond to the time after τ_{m-1} where we have recorded at least N'_m random evaluation points inside the ball $B(x^\star, 2 \cdot D \cdot 2^{-m})$. To prove the result, we show that each of the following events:

$$E_m := \left\{ \max_{i=1\dots\tau_m} f(X_i) \geq \max_{x \in \mathcal{X}} f(x) - \frac{c_\kappa}{2} \cdot \left(\frac{D}{2^m}\right)^\kappa \right\} \cap \left\{ \tau_m \leq N'_1 + \sum_{l=1}^{m-1} \left(N'_{l+1} + N_l\right) \right\}.$$

holds true with probability at least $1-\delta/M$ on the event $\bigcap_{l=1}^{m-1} E_l$ for all $m \in \{2,\dots,M\}$ so that:

$$\mathbb{P}(E_M) \geq \mathbb{P}(E_1) \times \prod_{m=1}^{M-1} \mathbb{P}\left(E_{m+1} | \bigcap_{l=1}^{m} E_l\right) \geq \left(1-\frac{\delta}{M}\right)^M \geq 1-\delta \tag{2.2}$$

and we will get to the desired result by analyzing E_M.

Analysis of $\mathbb{P}(E_1)$. Observe first that since $\mathcal{X} \subseteq B(x^\star, D)$, then $\tau_1 = N_1'$. Now, using the fact that (i) the algorithm is faster than a Pure Random Search (Proposition 2.13) and (ii) the bound of Proposition B.8, we directly get that with probability at least $1-\delta/M$,

$$\begin{aligned}
\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots\tau_1} f(X_i) &\leq k \cdot 2D \cdot \left(\frac{\ln(M/\delta)}{N_1'}\right)^{\frac{1}{d}} \\
&\leq k \cdot 2D \cdot \left(\frac{\ln(M/\delta)}{\ln(M/\delta)2^{d(\kappa-1)}} \left(\frac{c_\kappa D^\kappa}{8kD}\right)^d\right)^{\frac{1}{d}} \\
&= \frac{c_\kappa}{2} \cdot \left(\frac{D}{2}\right)^\kappa
\end{aligned}$$

which proves that $\mathbb{P}(E_1) \geq 1-\delta/M$.

Analysis of $\mathbb{P}(E_{m+1} | \cap_{l=1}^m E_l)$. To bound this term, we use (i) a deterministic covering argument to control the stopping time τ_{m+1} (Lemma 2.28 and Corollary 2.29) and (ii) a stochastic argument to bound the maximum $\max_{i=1\dots\tau_{m+1}} f(X_i)$ (Lemma 2.30 and Corollary 2.31). The following lemma states that after τ_m and on the event E_m there will be at most N_m evaluation points that will fall inside the area $B(x^\star, 2D\cdot 2^{-m})/B(x^\star, D\cdot 2^{-m})$.

Lemma 2.28. *For all $m \in \{1,\dots,M-1\}$, we have on the event E_m,*

$$\sum_{t=\tau_m+1}^{n} \mathbb{I}\left\{X_t \in B(x^\star, 2D\cdot 2^{-m})/B(x^\star, D\cdot 2^{-m})\right\} \leq N_m.$$

Proof. Fix $m \in \{1,\dots,M-1\}$ and assume that E_m holds true. Setting $N = \lceil \sqrt{d} 8kD 2^{m(\kappa-1)}/(c_k D^\kappa)\rceil$ and observing that $B(x^\star, 2D\cdot 2^{-m}) \subseteq x^\star + 2D\cdot 2^{-m}\cdot [-1,+1]^d$, one can then introduce the sequence H_I with $I \in \{1,\dots,N\}^d$ of the $N^d = N_m$ hypercubes whose side have length $4D\cdot 2^{-m}/N$ and cover $x^\star + 2D\cdot 2^{-m} \times [-1,+1]^d$, so that

$$B(x^\star, 2D\cdot 2^{-m})/B(x^\star, D\cdot 2^{-m}) \subseteq x^\star + 2D\cdot 2^{-m}\cdot[-1,+1]^d = \bigcup_I H_I.$$

Based on these hypercubes, one can define the set

$$I_t = \{I \in \{1,\dots,N\}^d : H_I \cap B(x^\star, 2D\cdot 2^{-m})/B(x^\star, D\cdot 2^{-m}) \cap \mathcal{X}_{k,t} \neq \emptyset\}$$

which contains the indexes of the hypercubes that still intersect the set of potential maximizers $\mathcal{X}_{k,t}$ at time t and the target area $B(x^\star, 2D\cdot 2^{-m})/B(x^\star, D\cdot 2^{-m})$. We show by

contradiction that there cannot be more than $N^d = N_m$ evaluation points falling inside this area, otherwise it would be empty. Suppose that, after τ_m, there exists a sequence

$$\tau_m < t_1 < t_2 < \cdots < t_{N^d+1} \leq n$$

of $N^d + 1$ strictly increasing indexes for which the evaluation points X_{t_j}, $j \geq 1$, belong to the target area, *i.e,*

$$\forall j \in \{1, \ldots, N^d + 1\},\ X_{t_j} \in B(x^\star, 2D \cdot 2^{-m}) / B(x^\star, D \cdot 2^{-m}).$$

Fix any $j \geq 1$ and observe that since $X_{t_j} \notin B(x^\star, D \cdot 2^{-m})$, then we have from Condition 2.16, (i) $f(X_{t_j}) < \max_{x \in \mathcal{X}} f(x) - c_k \cdot (D \cdot 2^{-m})^\kappa$. Moreover, as $X_{t_j} \in \mathcal{X}_{k,t_j-1} \cap B(x^\star, 2D \cdot 2^{-m}) / B(x^\star, D \cdot 2^{-m})$, it necessarily follows from the definition of the algorithm that (ii) there exists an index $I^\star \in I_{t_j-1}$ such that $X_{t_j} \in H_{I^\star}$. Therefore, combining (i) and (ii) with E_m, gives that $\forall x \in H_{I^\star}$:

$$\begin{aligned}
f(x) &\leq f(X_{t_j}) + k \cdot \left\| X_{t_j} - x \right\|_2 && (f \in \mathrm{Lip}(k)) \\
&\leq f(X_{t_j}) + k \cdot \max_{(x,x') \in H_I^2} \left\| x - x' \right\|_2 && ((X_{t_j}, x) \in H_I^2) \\
&= f(X_{t_j}) + k \cdot \sqrt{d} \cdot 4D \cdot 2^{-m} / N && (\text{def. of } H_I) \\
&\leq f(X_{t_j}) + \frac{c_k}{2} \cdot (D \cdot 2^{-m})^\kappa && (\text{def. of } N) \\
&< \max_{x \in \mathcal{X}} f(x) - c_\kappa \cdot (D \cdot 2^{-m})^\kappa + \frac{c_k}{2} \cdot (D \cdot 2^{-m})^\kappa && \text{(i)} \\
&\leq \max_{i=1 \ldots \tau_m} f(X_i) && (E_1) \\
&\leq \max_{i=1 \ldots t_j} f(X_i). && (t_j > \tau_m)
\end{aligned}$$

It has been shown that if X_{t_j} belongs to the target area, then $f(x) < \max_{i=1 \ldots t_j} f(X_i)$ for all $x \in H_{I^\star}$, which combined with the definition of the set of potential maximizers $\mathcal{X}_{k,t_j}$ at time t_j gives that $H_{I^\star} \notin \mathcal{X}_{\tau_j}$. Hence, once an evaluation has been made in $H_{I^\star}$, there will not be any future evaluation point falling inside this cube. We thus deduce that $|I_{t_j}| \leq |I_{t_j-1}| - 1$ for all $j \geq 1$ which leads us to the following contradiction:

$$0 \leq |I_{t_{N^d+1}}| = |I_{\tau_m}| + \sum_{j=\tau_m+1}^{t_{N^d+1}} |I_{t_j}| - |I_{t_{j-1}}| \leq |I_{\tau_m}| - (N^d + 1) \leq N^d - (N^d + 1) < 0$$

and proves the statement. □

Based on this lemma, one might then derive a bound on the stopping time τ_{m+1}.

Corollary 2.29. *For all $m \in \{1, \ldots, M-1\}$, we have on the event $\bigcap_{l=1}^m E_l$ that*

$$\tau_{m+1} \leq N_1' + \sum_{l=1}^{m} \left(N_{l+1}' + N_l \right).$$

Proof. The result is proved by induction. We start with the case where $m = 1$. Assuming that E_1 holds true and observing that (i) $\tau_1 = N_1'$, (ii) $\mathcal{X} \subseteq B(x^\star, D) = B(x^\star, D/2) \cup$

$B(x^\star, D)/B(x^\star, D/2)$ and (iii) for all $i \geq 1$, $X_i \in B(x^\star, D)/B(x^\star, D/2)$ or $X_i \in B(x^\star, D)$, one can then write:

$$\begin{aligned}\tau_2 &= \tau_1 + \sum_{i=\tau_1+1}^{\tau_2} \mathbb{I}\{X_i \in B(x^\star, D)\} \\ &= N_1' + \sum_{i=\tau_1+1}^{\tau_2} \mathbb{I}\{X_i \in B(x^\star, D/2)\} + \sum_{i=\tau_1+1}^{\tau_2} \mathbb{I}\{X_i \in B(x^\star, D)/B(x^\star, D/2)\}.\end{aligned}$$

However, since (i) $\sum_{i=\tau_1+1}^{\tau_2} \mathbb{I}\{X_i \in B(x^\star, D/2)\} = N_2'$ by definition of τ_2 and (ii) $\sum_{i=\tau_1+1}^{\tau_2} \mathbb{I}\{X_i \in B(x^\star, D)/B(x^\star, D/2)\} \leq N_1$ by Lemma 2.28, the result holds true for $m = 1$. Consider now any $m \geq 2$ and assume that the statement holds true for all $l < m$. Again, observing that $\mathcal{X} \subseteq B(x^\star, D \cdot 2^{-m}) \cup \bigcup_{l=1}^m B(x^\star, D \cdot 2^{-(l-1)})/B(x^\star, D \cdot 2^{-l})$ and keeping in mind that the stopping times are bounded by the induction assumption, one can then write:

$$\begin{aligned}\tau_{m+1} = \tau_m + \sum_{i=\tau_m+1}^{\tau_{m+1}} \mathbb{I}\left\{X_i \in B(x^\star, D \cdot 2^{-m})\right\} \\ + \sum_{i=\tau_m+1}^{\tau_{m+1}} \sum_{l=1}^{m} \mathbb{I}\left\{X_i \in B(x^\star, D \cdot 2^{-(l-1)})/B(x^\star, D \cdot 2^{-l})\right\}.\end{aligned}$$

Now, combining the telescopic representation $\tau_{m+1} = \tau_1 + \sum_{l=1}^m (\tau_{l+1} - \tau_l)$ with the previous decomposition gives that

$$\begin{aligned}\tau_{m+1} = \tau_1 + \sum_{l=1}^{m} \sum_{\tau_l+1}^{\tau_{l+1}} \mathbb{I}\left\{X_i \in B(x^\star, D \cdot 2^{-l})\right\} \\ + \sum_{l=1}^{m} \sum_{i=\tau_l+1}^{\tau_{m+1}} \mathbb{I}\left\{X_i \in B(x^\star, D \cdot 2^{-(l-1)})/B(x^\star, D \cdot 2^{-l})\right\}.\end{aligned}$$

However, since (i) $\tau_1 = N_1'$, (ii) $\sum_{\tau_l+1}^{\tau_{l+1}} \mathbb{I}\left\{X_i \in B(x^\star, D \cdot 2^{-l})\right\} = N_{l+1}'$, for all $l \geq 1$ by definition of the stopping times and (iii) $\sum_{i=\tau_l+1}^{\tau_{m+1}} \mathbb{I}\left\{X_i \in B(x^\star, 2^{-(l-1)})/B(x^\star, 2^{-l})\right\} \leq N_l$ for all $l \geq 1$ on the event $\bigcap_{l=1}^m E_l$ from Lemma 2.28, we finally get that

$$\tau_{m+1} \leq N_1' + \sum_{l=1}^{m} (N_{l+1}' + N_l).$$

□

Since Corollary 2.29 gives us the desired bound on τ_{m+1}, it remains to control the maximum $\max_{i=1\ldots\tau_{m+1}} f(X_i)$. The next lemma shows that i.i.d. results can actually be used to bound this term.

Lemma 2.30. *For all $m \in \{1, \ldots, M-1\}$, we have that $\forall y \in \mathrm{Im}(f)$,*

$$\mathbb{P}\left(\max_{i=1\ldots\tau_{m+1}} f(X_i) \geq y \mid \bigcap_{l=1}^{m} E_l\right) \geq \mathbb{P}\left(\max_{i=1\ldots N_{m+1}'} f(X_i') \geq y\right).$$

where $X_1' \ldots X_{N_{m+1}'}'$ denotes a sequence N_{m+1}' i.i.d. copies of $X' \sim \mathcal{U}(\mathcal{X} \cap B(x^\star, D \cdot 2^{-m}))$.

Proof. From Corollary 2.29, we know that on the event $\bigcap_{l=1}^{m} E_l$ the stopping time τ_{m+1} is finite. Moreover, as $\sum_{i=\tau_m+1}^{\tau_{m+1}} \mathbb{I}\{X_i \in B(x^\star, D\cdot 2^{-m})\} = N'_{m+1}$ by definition of τ_{m+1}, it can then easily be shown by reproducing the same steps as in the proof of Proposition 2.13 with the evaluations points falling into $B(x^\star, D\cdot 2^{-m})$ after τ_m that the algorithm is faster than a Pure Random Search in the usual stochastic ordering sense performed over the domain $\mathcal{X} \cap B(x^\star, D\cdot 2^{-m})$, which proves the result. □

As a direct consequence of this lemma, one can get the desired bound on the maxima, as shown in the next corollary.

Corollary 2.31. *For all $m \in \{1,\dots,M-1\}$, we have that*

$$\mathbb{P}\left(\max_{i=1\dots\tau_{m+1}} f(X_i) \geq \max_{x\in\mathcal{X}} f(x) - \frac{c_k}{2}\cdot\left(\frac{D}{2^{m+1}}\right)^{\kappa} \mid \bigcap_{l=1}^{m} E_l\right) \geq 1-\delta/M.$$

Proof. Omitting the conditioning upon $\bigcap_{l=1}^{m} E_l$, we obtain from the combination of Lemma 2.30 and Proposition B.8 that with probability at least $1-\delta/M$:

$$\begin{aligned}
\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots\tau_{m+1}} f(X_i) &\leq k\cdot 2D\cdot 2^{-m}\cdot\left(\frac{\ln(M/\delta)}{N'_{m+1}}\right)^{\frac{1}{d}} \\
&\leq k\cdot 2D\cdot 2^{-m}\cdot\left(\frac{\ln(M/\delta)}{\ln(M/\delta)2^{d(m+1)(\kappa-1)}}\left(\frac{c_\kappa D^\kappa}{8kD}\right)^{d}\right)^{\frac{1}{d}} \\
&= \frac{c_\kappa}{2}\cdot\left(\frac{D}{2^{m+1}}\right)^{\kappa}.
\end{aligned}$$

□

Finally, putting all the previous results altogether, we know at this point from the combination of Corollary 2.29 and Corollary 2.31 that

$$\forall m\in\{1,\dots,M-1\},\ \mathbb{P}\left(E_{m+1}|\bigcap_{l=1}^{m} E_l\right) \geq 1-\delta/M$$

which proves that $\mathbb{P}(E_M)\geq 1-\delta$ from (2.2).

Analysis of E_M. Since $\max_{i=1\dots\tau_M} f(X_i) \geq \max_{x\in\mathcal{X}} f(x) - \frac{c_\kappa}{2}\cdot D^\kappa\cdot 2^{-M\kappa}$ and $\tau_M \leq N'_1 + \sum_{l=1}^{M-1}\left(N'_{l+1}+N_l\right)$ on the event E_M, it remains to show that $N'_1 + \sum_{l=1}^{M-1}\left(N'_{l+1}+N_l\right) \leq n$ to conclude the proof. Consider the case $\kappa=1$. Setting $C = (8k/c_\kappa)^d$ and observing that (i) $N'_l \leq \ln(M/\delta)C+1$, (ii) $N_l \leq 2\cdot C\cdot(2\sqrt{d})^d - 1$ for all $l\leq M$ and (iii) $M\leq n$, one gets:

$$\begin{aligned}
N'_1 + \sum_{l=1}^{M-1}\left(N'_{l+1}+N_l\right) &\leq C\cdot M\left(\ln(M/\delta) + 2(2\sqrt{d})^d\right) \\
&\leq n\cdot\frac{\ln(M/\delta)+2(2\sqrt{d})^d}{\ln(n/\delta)+2(2\sqrt{d})^d} \\
&\leq n.
\end{aligned}$$

Now, considering the case $\kappa > 1$ and since (i) M was chosen so that $\frac{2^{d(\kappa-1)M}-1}{2^{d(\kappa-1)}-1} \leq \frac{n}{C} \cdot \frac{1}{\ln(n/\delta)+2(2\sqrt{d})^d}$ and (ii) $M \leq n$, we obtain:

$$\begin{aligned} N_1' + \sum_{l=1}^{M-1} \left(N_{l+1}' + N_l\right) &\leq C \cdot \left(\ln(M/\delta) + 2(2\sqrt{d})^d\right) \sum_{l=1}^{M} (2^{d(\kappa-1)})^l \\ &\leq C \cdot \left(\ln(M/\delta) + 2(2\sqrt{d})^d\right) \cdot \frac{2^{d(\kappa-1)M}-1}{2^{d(\kappa-1)}-1} \\ &\leq n. \end{aligned}$$

Finally, using the elementary inequality $\lfloor x \rfloor \geq x - 1$ on M and the inequality $c_\kappa D^\kappa \leq k\,\mathrm{diam}(\mathcal{X})$ (Condition 2.16) leads to the desired result and completes the proof. □

Proof of Theorem 2.18. (Lower bound) This result mostly rely on the fact that the algorithm is slower than a Pure Adaptive Search (see Definition 3.17 and Proposition 3.18 in Chapter 3). Pick any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, set $\epsilon = c_\kappa \mathrm{rad}(\mathcal{X})^\kappa \delta^{\kappa/d} \exp(-\kappa(n - \sqrt{2n\ln(1/\delta)})/d)$, let $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$ be the corresponding level set. Observe first that since (i) $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : \epsilon \geq f(x^\star) - f(x)\} \subseteq \{x \in \mathcal{X} : \epsilon \geq c_\kappa \|x^\star - x\|_2^\kappa\} = \mathcal{X} \cap B(x^\star, (\epsilon/c_\kappa)^{1/\kappa}$ and (ii) there exists $x \in \mathcal{X}$ such that $B(x, \mathrm{rad}(\mathcal{X})) \subseteq \mathcal{X}$, then $\mu(\mathcal{X}_\epsilon)/\mu(\mathcal{X}) \leq ((\epsilon/c_\kappa)^{1/\kappa}/\mathrm{rad}(\mathcal{X}))^d = \delta e^{-n-\sqrt{2n\ln(1/\delta)}}$. One can then easily show by reproducing the same steps as in the proof of the Lower bound of Theorem 3.19 in Chapter 3 that

$$\begin{aligned} \mathbb{P}\left(\max_{i=1\ldots n} f(X_i) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\right) &= \mathbb{P}\left(\frac{\mu(\{x \in \mathcal{X} : f(x) \geq \max_{i=1\ldots n} f(X_i)\})}{\mu(\mathcal{X})} \leq \frac{\mu(\mathcal{X}_\epsilon)}{\mu(\mathcal{X})}\right) \\ &\leq \mathbb{P}\left(\prod_{i=1}^{n} U_i \leq \frac{\mu(\mathcal{X}_\epsilon)}{\mu(\mathcal{X})}\right) \\ &\leq \mathbb{P}\left(\prod_{i=1}^{n} U_i \leq \delta \cdot e^{-n-\sqrt{2n\ln(1/\delta)}}\right) \\ &= \mathbb{P}\left(\sum_{i=1}^{n} -\ln(U_i) > n + \sqrt{2n\ln(1/\delta)} + \ln(1/\delta)\right) \\ &\leq \delta \end{aligned}$$

where $U_1, \ldots, U_n$ denotes a sequence of n i.i.d. copies of $U \sim \mathcal{U}([0,1])$ and proves the result. We point out that a concentration results for gamma random variable was used on the last line (see Lemma 3.37 and Lemma 3.38 in the next Chapter for more details). □

2.6.3 Proofs of Section 2.4

Proof of Proposition 2.21. Pick any $t \geq 2$, consider any non-constant $f \in \bigcup_{k\geq 0} \mathrm{Lip}(k)$ and set $i^\star = \min\{i \in \mathbb{Z} : f \in \mathrm{Lip}(k_i)\}$. To prove the result, we decorrelate the sample and use the fact that $(X_1, X_{\lfloor t/2 \rfloor + 1}), \ldots, (X_{\lfloor t/2 \rfloor}, X_{2\lfloor t/2 \rfloor})$ forms a sequence of $\lfloor t/2 \rfloor$ i.i.d. copies

of $(X, X') \sim \mathcal{U}(\mathcal{X} \times \mathcal{X})$:

$$\begin{aligned}
\mathbb{P}\left(f \in \mathrm{Lip}(\hat{k}_t)\right) &= \mathbb{P}\left(\hat{k}_t = k_{i^\star}\right) \\
&= \mathbb{P}\left(\bigcup_{i \neq j}^{t} \left\{ \left|f(X_i) - f(X_j)\right| > k_{i^\star - 1} \cdot \left\|X_i - X_j\right\|_2 \right\}\right) \\
&\geq \mathbb{P}\left(\bigcup_{i=1}^{\lfloor t/2 \rfloor} \left\{ \left|f(X_i) - f(X_{\lfloor t/2 \rfloor + i})\right| > k_{i^\star - 1} \cdot \left\|X_i - X_{\lfloor t/2 \rfloor + i}\right\|_2 \right\}\right) \\
&= 1 - \mathbb{P}\left(\frac{|f(X_1) - f(X_2)|}{\|X_1 - X_2\|_2} \leq k_{i^\star - 1}\right)^{\lfloor t/2 \rfloor} \\
&= 1 - \left(1 - \Gamma(f, k_{i^\star - 1})\right)^{\lfloor t/2 \rfloor}.
\end{aligned}$$

It remains to show that $\Gamma(f, k_{i^\star - 1}) > 0$ to conclude the proof. Observe first that since $f \in \mathrm{Lip}(k_{i^\star})$, then the function $F : (x, x') \mapsto |f(x) - f(x')| - k_{i^\star - 1} \cdot \|x - x'\|_2$ is also continuous. However, as $f \notin \mathrm{Lip}(k_{i^\star - 1})$, we know that there exists some $(x_1, x_2) \in \mathcal{X} \times \mathcal{X}$ such that $F(x_1, x_2) > 0$. Hence, it follows from the continuity of F that there also exists some $\epsilon > 0$ such that $\forall (x, x') \in B(x_1, \epsilon) \cap \mathcal{X} \times B(x_2, \epsilon) \cap \mathcal{X}$, $F(x, x') > 0$ which proves the result. □

Proof of Proposition 2.25. Combining the consistency equivalence of Proposition 2.3 with the upper bound on the covering rate obtained in Example 2.4 gives the result. □

Proof of Proposition 2.26. Fix any $\delta \in (0,1)$, set $N_1 = 2 + \lceil 2\ln(\delta/3)/\ln(1 - \Gamma(f, k_{i^\star - 1}))\rceil$ and set $N_2 = \lceil ((\sqrt{\ln(3/\delta)/2 + 4N_1 p} - \sqrt{\ln(3/\delta)/2})/2p)^2 \rceil$. Consider any $n > N_2$. We prove the result in three steps.

Step 1. As the constant N_1 and N_2 were chosen so that Hoeffding's inequality ensures that $\mathbb{P}\left(\sum_{i=1}^{N_2} B_i \geq N_1\right) \geq 1 - \delta/3$, we know that after N_2 iterations and with probability $1 - \delta/3$, we have collected at least N_1 evaluation points randomly and uniformly distributed over $\mathcal{X}$ due to the exploration step.

Step 2. Using Proposition 2.21 and the fist N_1 evaluation points that have been sampled independently and uniformly over $\mathcal{X}$, we know that after N_2 iterations and on the event $\{\sum_{i=1}^{N_2} B_i \geq N_1\}$ the Lipschtz constant $k_{i^\star}$ has been estimated with probability at least $1 - \delta/3$, *i.e.*, $\mathbb{P}\left(\forall t \geq N_2 + 1,\, \hat{k}_t = k_{i^\star} \mid \sum_{i=1}^{N_2} B_i \geq N_1\right) \geq 1 - \delta/3$.

Step 3. Finally, as the Lipschitz constant estimate $\hat{k}_t$ satisfies $f \in \mathrm{Lip}(\hat{k}_t)$ for all $t \geq N_2 + 1$ on the above event, one can easily show by reproducing the same steps as in Proposition 2.13 that conditioned upon the event $\{\forall t \geq N_2 + 1,\, \hat{k}_t = k_{i^\star}\} \cap \{\sum_{i=1}^{N_2} B_i \geq N_1\}$ the algorithm is always faster or equal to a Pure Random Search ran with $n - N_2$ i.i.d. copies of $X' \sim \mathcal{U}(\mathcal{X})$. Therefore, using (i) the bound of Proposition B.8, (ii) the elementary inequalities $\lceil x \rceil \leq x + 1$, $\lfloor x \rfloor \geq x - 1$, $\sqrt{x + y} - \sqrt{x} \leq \sqrt{y}$ and (iv) the definition of

$N_2 < n$, we obtain that with probability at least $(1-\delta/3)^3 \geq 1-\delta$,

$$\begin{aligned}
\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) &\leq k_{i^\star} \cdot \operatorname{diam}(\mathcal{X}) \cdot \left(\frac{\ln(3/\delta)}{n-N_2}\right)^{\frac{1}{d}} \\
&= k_{i^\star} \cdot \operatorname{diam}(\mathcal{X}) \cdot \left(\frac{n}{n-N_2}\right)^{\frac{1}{d}} \cdot \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}} \\
&\leq k_{i^\star} \cdot \operatorname{diam}(\mathcal{X}) \cdot (1+N_2)^{\frac{1}{d}} \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}} \\
&\leq k_{i^\star} \cdot \operatorname{diam}(\mathcal{X}) \cdot \left(\frac{5}{p} + \frac{2\ln(\delta/3)}{p\ln(1-\Gamma(f,k_{i^\star-1}))}\right)^{\frac{1}{d}} \cdot \left(\frac{\ln(3/\delta)}{n}\right)^{\frac{1}{d}}.
\end{aligned}$$

The result is then extended to the case where $n \leq N_2$ by noticing that the bound is superior to $k_{i^\star} \cdot \operatorname{diam}(\mathcal{X})$ in that case, and thus trivial. □

Proof of Proposition 2.27. Fix $\delta \in (0,1)$, set $N_1 = 2 + \lceil 2\ln(4/\delta)/\ln(1-\Gamma)\rceil$, $N_2 = \lceil((\sqrt{\ln(4/\delta)/2 + 4N_1 p} - \sqrt{\ln(4/\delta)/2})/2p)^2\rceil$ and let $N_3 = N_2 + \lceil 2\ln(4/\delta)/(1-p)^2\rceil$. Picking any $n > N_3$, we proceed in four steps, similarly as in the proof of Proposition 2.26:

Steps 1 & 2. As in the above proof, by definition of N_1 and N_2, by Hoeffding's inequality and by Proposition 2.21, we know that the following event: $\{\forall t \geq N_2+1, \hat{k}_t = k_{i^\star}\} \cap \{\sum_{i=1}^{N_2} B_i \geq N_1\}$ holds true with probability at least $(1-\delta/4)^2$.

Step 3. Again, using Hoeffding's inequality and the definition of N_2 and N_3, we know that after the iteration N_2+1 we have collected with probability at least $1-\delta/4$ at least $(1-p)(n-N_3)/2$ exploitative evaluation points:

$$\sum_{i=N_2+1}^{n} \mathbb{I}\{B_i = 0\} \geq (1-p)(n-N_2) - \sqrt{\frac{(n-N_2)\ln(4/\delta)}{2}} \geq \frac{1-p}{2}\cdot(n-N_3).$$

Step 4. Reproducing the same steps as in the proof of the fast rate of Theorem 2.17 with the $(1-p)\cdot(n-N_3)/2$ previous exploitative points and putting the previous results altogether gives that with probability at least $(1-\delta/4)^4 \geq 1-\delta$,

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k_{i^\star} \times \operatorname{diam}(\mathcal{X}) \times$$

$$\begin{cases}
\exp\left\{-C_{k,\kappa}\cdot\dfrac{(1-p)(n-N_3)\ln(2)}{2\ln(4n/\delta)+4(2\sqrt{d})^d}\right\}, & \kappa = 1 \\[2ex]
\dfrac{2^\kappa}{2}\left(1 + C_{k,\kappa}\cdot\dfrac{(1-p)(n-N_3)(2^{d(\kappa-1)}-1)}{2\ln(4n/\delta)+4(2\sqrt{d})^d}\right)^{-\frac{\kappa}{d(\kappa-1)}}, & \kappa > 1.
\end{cases}$$

We now simplify the convergence rate. Since $C_{k_{i^\star},\kappa}(1-p) \leq 1$ when $\kappa = 1$, we directly have

$$\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq k_{i^\star} \times \operatorname{diam}(\mathcal{X}) \times \exp(5N_3/2) \times \exp\left\{ - C_{k,\kappa} \cdot \frac{(1-p)n\ln(2)}{2\ln(4n/\delta) + 4(2\sqrt{d})^d} \right\}.$$

For $\kappa > 1$, setting $C = C_{k_{i^\star},\kappa}(1-p)/(2\ln(4n/\delta) + 4(2\sqrt{d})^d)$ and using the decomposition $n = (n - N_3) + N_3$, we bound the ratio as follows:

$$\left(\frac{1 + Cn(2^{d(\kappa-1}-1)}{1 + C(n-N_3)(2^{d(\kappa-1)}-1)} \right)^{\frac{\kappa}{d(\kappa-1)}} \leq \left(1 + \frac{CN_3(2^{d(\kappa-1)}-1)}{1 + C(2^{d(\kappa-1)}-1)} \right)^{\frac{\kappa}{d(\kappa-1)}}.$$

In the case where $\kappa/d(\kappa-1) \leq 1$, one directly obtains

$$\left(1 + \frac{CN_3(2^{d(\kappa-1)}-1)}{1 + C(2^{d(\kappa-1)}-1)} \right)^{\frac{\kappa}{d(\kappa-1)}} \leq (1+N_3)^{\frac{\kappa}{d(\kappa-1)}} \leq (1+N_3) \leq e^{N_3}$$

Now, considering that $\kappa/d(\kappa-1) > 1$ and setting $\kappa = 1 + \epsilon/d$ with $\epsilon \in (0,1)$, we obtain from the inequalities (i) $\kappa \leq 1 + 1/d \leq 2$, (ii) $\forall \epsilon \in (0,1)$, $2^\epsilon - 1 \leq \epsilon$ and (iii) $C \leq 1/2$ that

$$\left(1 + \frac{CN_3(2^{d(\kappa-1)}-1)}{1 + C(2^{d(\kappa-1)}-1)} \right)^{\frac{\kappa}{d(\kappa-1)}} \leq (1 + CN_3(2^\epsilon - 1))^{\frac{2}{\epsilon}} \leq (1 + CN_3\epsilon)^{\frac{2}{\epsilon}} \leq e^{2CN_3} \leq e^{N_3}.$$

Finally, using standard bounds on N_3 and noticing that the previous bound is superior to $k_{i^\star}\operatorname{diam}(\mathcal{X})$ whenever $n \leq N_3$, the previous result remains valid for any $n \in \mathbb{N}^\star$. □

3

Non-smooth optimization and ranking

Abstract. The purpose of this chapter is to design sequential optimization strategies that do not rely on the smoothness of the unknown function. We observe that the optimization of the function essentially relies on learning the bipartite ranking rule it defines. Based on this idea, we relate global optimization to bipartite ranking which allows to address problems with high dimensional input space, as well as cases of functions with weak regularity properties. This chapter introduces novel meta-algorithms for global optimization which rely on the choice of any bipartite ranking method. Theoretical properties are provided as well as convergence guarantees and equivalences between various optimization methods are obtained as a byproduct. A significant part of this work has been published in Malherbe et al. (2016).

Contents

3.1 Introduction

This chapter follows the line of the approaches recently considered in the machine learning literature (Bull (2011); Sergeyev et al. (2013); Munos (2014); Malherbe and Vayatis (2017)). These approaches extend the seminal work on Lipschitz optimization of Hansen et al. (1992) and Jones et al. (1993) and they led to significant relaxations of the conditions required for convergence, *e.g.*, only the existence of a local smoothness around the optimum is required (Munos (2014); Grill et al. (2015)). Precisely, in the works of Bull (2011), Munos (2014) and Malherbe and Vayatis (2017), specific conditions have been identified to derive a finite-time analysis of the algorithms. However, these guarantees do not hold whenever the unknown function is not assumed to be locally smooth around (one of) its optimum. In the present chapter, we propose to explore concepts from ranking theory based on overlaying estimated level sets (Clémençon et al. (2010)) in order to develop global optimization algorithms that do not rely on the smoothness of the function. The idea behind this approach is simple: even if the unknown function presents arbitrary large variations, most of the information required to identify its optimum may be contained in its induced ranking rule, *i.e.* how the level sets of the function are included one in another. To exploit this idea, we introduce a novel optimization scheme where the complexity of the function is characterized by the underlying pairwise ranking which it defines. Our contribution is twofold: first, we introduce two novel global optimization algorithms that learn the ranking rule induced by the unknown function with a sequential scheme, and second, we provide mathematical results in terms of statistical consistency and convergence to the optimum. Moreover, the algorithms proposed lead to efficient implementation and display good performance on the classical benchmarks for global optimization, as shown in the Chapter 4. The remainder of this chapter is structured as follows. In Section 3.2, we recall the framework and introduce the main definitions. In Section 3.3, we introduce and analyze the RankOpt algorithm which requires the knowledge of a ranking structure underlying the unknown function. In Section 3.4, an adaptive version of the algorithm is presented. Finally, companion results which establish the equivalence between learning algorithms and optimization procedures are discussed in Section 3.5 as they support implementation choices. All proofs are postponed to the Section 3.7 and an emprical comparison of the adaptive version of the algorithm to existing methods can be found in Chapter 4.

3.2 Global optimization and ranking structure

3.2.1 Setup and notations

Setup. Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact and convex set and let $f : \mathcal{X} \to \mathbb{R}$ be an unknown function which is only supposed to admit a global maximum over its domain $\mathcal{X}$. The goal in global optimization consists in finding some point

$$x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$$

with a minimal amount of function evaluations. The standard setup involves a sequential procedure which starts by evaluating the function at a random initial point $X_1 \in \mathcal{X}$ and then iteratively selects at each step $t \geq 1$ a random evaluation point $X_{t+1} \in \mathcal{X}$ which depends on the previous evaluations $(X_1, f(X_1), \ldots, (X_t, f(X_t))$ and is $\mathcal{F}_t$-measurable

where $\mathcal{F}_t = \sigma((X_1, f(X_1)), \ldots, (X_t, f(X_t)))$ is the filtration generated by the history of available information, and recieves the evaluation of the unknown function $f(X_{t+1})$ at this point. After n iterations, we consider that the algorithm returns the argument of the highest evaluation observed so far:

$$X_{\hat{\imath}_n} \quad \text{where} \quad \hat{\imath}_n \in \underset{i=1\ldots n}{\arg\max}\, f(X_i).$$

The analysis provided here considers that the number n of evaluation points is not fixed and it is assumed that function evaluations are noiseless.

Notations. For all $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we define the standard ℓ_2-norm as $\|x\|_2^2 = \sum_{i=1}^d x_i^2$, we denote by $\langle \cdot, \cdot \rangle$ the corresponding inner product and we denote by $B(x,r) = \{x' \in \mathbb{R}^d : \|x - x'\|_2 \leq r\}$ the ℓ_2-ball centered in x of radius $r \geq 0$. For any bounded set $\mathcal{X} \subset \mathbb{R}^d$, we define its inner-radius as $\mathrm{rad}(\mathcal{X}) = \max\{r > 0 : \exists x \in \mathcal{X} \text{ such that } B(x,r) \subseteq \mathcal{X}\}$, its diameter as $\mathrm{diam}(\mathcal{X}) = \max_{(x,x') \in \mathcal{X}^2} \|x - x'\|_2$ and we denote by $\mu(\mathcal{X})$ its volume where μ stands for the Lebesgue measure. We denote by $\mathcal{C}^0(\mathcal{X}, \mathbb{R})$ the set of continuous functions defined on $\mathcal{X}$ taking values in $\mathbb{R}$, we denote by $\mathcal{P}_k(\mathcal{X}, \mathbb{R})$ the set of (multivariate) polynomial functions of degree $k \geq 1$ defined on $\mathcal{X}$, and for any function $f : \mathcal{X} \to \mathbb{R}$, we denote by $\mathrm{Im}(f) = \{f(x) : x \in \mathcal{X}\}$ its image. Finally, we denote by $\mathcal{U}(\mathcal{X})$ the uniform distribution over a bounded measurable domain $\mathcal{X}$, we denote by $\mathbb{I}\{\cdot\}$ the indicator function taking values in $\{0,1\}$ and we denote by $\mathrm{sgn}(\cdot)$ the standard sign function defined on $\mathbb{R}$ and taking values in $\{-1,0,1\}$.

3.2.2 The ranking structure of a real-valued function

In this section, we introduce the ranking structure as a characterization of the complexity for a general real-valued function to be optimized. First, we observe that real-valued functions induce an order relation over the input space $\mathcal{X}$, and the underlying ordering induces a ranking rule which records pairwise comparisons between evaluation points.

Definition 3.1. (Induced ranking rule) *The ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1,0,1\}$ induced by a function $f : \mathcal{X} \to \mathbb{R}$ is defined by:*

$$r_f(x,x') = \begin{cases} +1 & \text{if } f(x) > f(x') \\ 0 & \text{if } f(x) = f(x') \\ -1 & \text{if } f(x) < f(x') \end{cases}$$

for all $(x,x') \in \mathcal{X}^2$.

The key argument of our approach is that the optimization of any weakly regular real-valued function only depends on the nested structure of its level sets. Hence there is an equivalence class of real-valued functions that share the same induced ranking rule as shown by the following proposition.

Proposition 3.2. (Ranking rule equivalence) *Let $h \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$ be any continuous function. Then, a function $f : \mathcal{X} \to \mathbb{R}$ shares the same induced ranking rule with h (i.e. $\forall (x,x') \in \mathcal{X}^2$, $r_f(x,x') = r_h(x,x')$) if and only if there exists a strictly increasing, but not necessarily continuous function $\psi : \mathbb{R} \to \mathbb{R}$ such that $h = \psi \circ f$.*

The previous proposition states that even if the unknown function f admits noncontinuous or large variations, up to a transformation ψ, there might exist a simpler function $h = \psi \circ f$ that shares the same induced ranking rule. Figure 3.1 gives an example of three functions that share the same ranking rule while they display highly different regularity properties. As a second example, we may consider the problem of maximizing the function $f(x) = 1 - 1/|\ln(x)|$ if $x \neq 0$ and 1 otherwise over $\mathcal{X} = [0, 1/2]$. In this case, the unknown function f is not 'smooth' around its unique global maximizer $x^\star = 0$ but shares the same induced ranking rule with $h(x) = -x$ over $\mathcal{X}$.

A ranking structure is a collection of ranking rules. The approach developed in this chapter consists of seeing the ranking structure as a characterization of the complexity of the target function f through the complexity of its induced ranking rule. We first introduce a very large class of ranking rules.

Definition 3.3. (Continuous Ranking Structure and Continuous Ranking Rules) *The set of continuous ranking rules (i.e. the set of ranking rules induced by continuous functions) is defined as follows* $\mathcal{R}_\infty := \{r_h \mid h \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})\}$. *In addition, we say that a real-valued function* $f : \mathcal{X} \to \mathbb{R}$ *has a continuous ranking rule if* $r_f \in \mathcal{R}_\infty$.

In the continuation of this definition, we further introduce three examples of more stringent ranking structures.

Example 3.4. (Polynomial Ranking Rules) *The set of polynomial ranking rules of degree* $k \geq 1$ *is defined as*

$$\mathcal{R}_{\mathcal{P}_k} := \{r_h : (x, x') \mapsto \operatorname{sgn}(h(x) - h(x')) \mid h \in \mathcal{P}_k(\mathcal{X}, \mathbb{R})\}.$$

We point out that even a polynomial function of degree $k > 1$ may admit a lower degree polynomial ranking rule. For example, consider the polynomial function $f(x) = (x^2 - 3x + 1)^9$. Since $f(x) = \psi(x^2 - 3x)$ where $\psi : x \mapsto (x+1)^9$ is a strictly increasing function, the ranking rule induced by f is a polynomial ranking rule of (at most) degree 2. We may now introduce our second class of ranking structure which is a direct extension of the class of polynomial ranking rules.

Example 3.5. (Sinusoidal Ranking Rules) *The set of sinusoidal ranking rules of degree* $k \geq 1$ *is defined as*

$$\mathcal{R}_{\mathcal{S}_k} := \{r_h : (x, x') \mapsto \operatorname{sgn}\big(\big(h(\cos(2\pi x)) - h(\cos(2\pi x'))\big)\big) \mid h \in \mathcal{P}_k(\mathcal{X}, \mathbb{R})\}$$

where the cosine function is vectorized, i.e. $\forall x \in \mathbb{R}^d$, $cos(x) = \{cos(x_1), \ldots, cos(x_d)\}$.

The last class of ranking structures we introduce is a class of non-parametric ranking rules.

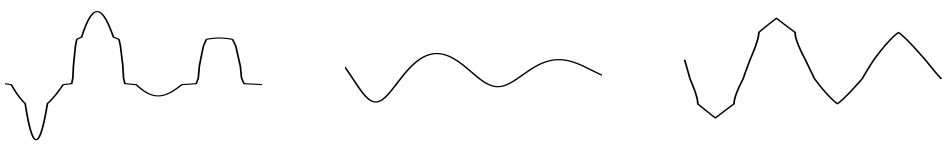


Figure 3.1: Three functions that share the same ranking rule

Example 3.6. (Convex Ranking Rules) *The set of convex ranking rules of degree $k \geq 1$ is defined as*

$$\mathcal{R}_{\mathcal{C}_k} := \{r \in \mathcal{R}_\infty \text{ such that for any } x' \in \mathcal{X}, \text{ the set } \{x \in \mathcal{X} : r(x,x') \geq 0\} \text{ is a union of } k \text{ convex sets}\}.$$

It is easy to see that the ranking rule of a function f is a convex ranking rule of degree k if and only all the level sets of the function f are unions of at most k convex sets.

3.2.3 Identifiability and regularity

We state here two conditions that will be used in the theoretical analysis: the first condition is about the identifiability of the maximum of the function and the second is about the regularity of the function around its maximum.

Condition 3.7. (Identifiability) *The maximum of a function $f : \mathcal{X} \to \mathbb{R}$ is said to be identifiable if for any $\varepsilon > 0$ arbitrarily small,*

$$\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x \in \mathcal{X}} f(x) - \varepsilon\}) > 0.$$

This condition prevents the function from having a spike on its maximum. It will be useful to state asymptotic results of the type $\max_{i=1\ldots n} f(X_i) \to \max_{x \in \mathcal{X}} f(x)$ when $n \to +\infty$.

Condition 3.8. (Regularity of the level sets) *A function $f : \mathcal{X} \to \mathbb{R}$ has (c_α, α)-regular level sets for some $c_\alpha > 0$, $\alpha \geq 0$ if:*

1. *The global optimizer $x^\star \in \mathcal{X}$ is unique;*
2. *For any $y \in \mathrm{Im}(f)$, the iso-level set $f^{-1}(y) = \{x \in \mathcal{X} : f(x) = y\}$ satisfies*

$$\max_{x \in f^{-1}(y)} \|x^\star - x\|_2 \leq c_\alpha \cdot \min_{x \in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)}.$$

Condition 3.8 guarantees that the points associated with high evaluations are close to the unique optimizer with respect to the Euclidean distance. Note however that for any iso-level set $f^{-1}(y)$ with finite distance to the optimum, the condition is satisfied with $\alpha = 0$ and $c_\alpha = \mathrm{diam}(\mathcal{X}) / \min_{x \in f^{-1}(y)} \|x^\star - x\|_2$. Thus, this condition concerns the local behavior of the level sets when $\min_{x \in f^{-1}(y)} \|x^\star - x\|_2 \to 0$. As an example, the iso-level sets of three simple functions satisfying the condition with different values of α are shown in Figure 3.2.

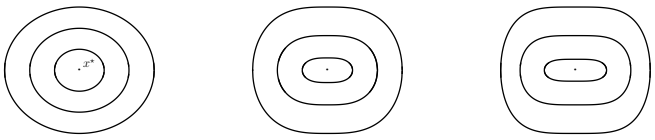


Figure 3.2: Illustration of the regularity of the level sets on three simple functions. *Left:* $f(x_1, x_2) = -x_1^2 - 1.4x_2^2$ where $\alpha = 0$. *Middle:* $f(x_1, x_2) = \exp(-|x_1|^3 - 1.4x_2^2)$ where $\alpha = 1/2$. *Right:* $f(x_1, x_2) = -x_1^4 - 1.4x_2^2$ where $\alpha = 1$.

3.3 Optimization with fixed ranking structure

In this section, we consider the problem of optimizing an unknown function f given the knowledge that its induced ranking rule r_f belongs to a given ranking structure $\mathcal{R} \subseteq \mathcal{R}_\infty$.

3.3.1 The RankOpt algorithm

Definitions. To properly set up the algorithm, we first introduce some key concepts that will used throughout the chapter. We start with the definition of the empirical ranking loss.

Definition 3.9. (Empirical ranking loss) *The empirical ranking loss computed over a sample $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ of $t \geq 2$ function evaluations is defined for all $r : \mathcal{X} \times \mathcal{X} \to \{-1, 0, 1\}$ by*

$$L_t(r) := \frac{2}{t(t-1)} \sum_{1 \leq i < j \leq t} \mathbb{I}\big\{r(X_i, X_j) \neq r_f(X_i, X_j)\big\}$$

where $r_f(X_i, X_j) = \operatorname{sgn}(f(X_i) - f(X_j))$ for all $(i, j) \in \{1, \ldots, t\}^2$.

Based on this definition, one might then recover among a ranking structure $\mathcal{R}$ the subset of ranking rules r which are consistent with the ranking rule r_f induced by the function over a sample of function evaluations.

Definition 3.10. (Active subset of consistent ranking rules) *The active subset of a ranking structure $\mathcal{R}$ which contains the ranking rules consistent with r_f over a sample $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ of $t \geq 2$ function evaluations is defined by*

$$\mathcal{R}_t := \{r \in \mathcal{R} : L_t(r) = 0\}$$

where $L_t(\cdot)$ denotes the empirical ranking loss defined above.

We are now ready to introduce the optimization algorithm.

Algorithm description. The input of the RankOpt algorithm (displayed in Figure 3.3) are a number n of iterations, a ranking structure $\mathcal{R} \subseteq \mathcal{R}_\infty$, a compact and convex set $\mathcal{X} \subset \mathbb{R}^d$ and the target function f. At each iteration $t < n$, a point X_{t+1} is sampled uniformly over $\mathcal{X}$ and the algorithm decides, whether or not, to evaluate the function at this point. The decision rule involves the active subset $\mathcal{R}_t$ which contains the ranking rules that are consistent with the ranking rule induced by f over the points sampled so far. More precisely, the decision rule operates as follows: if there does not exist any ranking rule $r \in \mathcal{R}_t$ which satisfies $r(X_{t+1}, X_{\hat{\imath}_t}) \geq 0$, then we know from the definition of $\mathcal{R}_t$ that $r_f(X_{t+1}, X_{\hat{\imath}_t}) = -1$ which necessarily means that $f(X_{t+1}) < f(X_{\hat{\imath}_t})$. Thus, the algorithm never evaluates the function at a point that will not return certainly an evaluation at least equal to the highest evaluation $f(X_{\hat{\imath}_t})$ observed so far.

Remark 3.11. (Connection with active learning) *Although the problem considered in this chapter is very different, the RankOpt algorithm might be seen as an extension to ranking of the baseline active learning algorithm introduced in Cohn et al. (1994) and further analyzed by Hanneke (2011). However, the main difference with this algorithm lies in the fact*

Input: $n \in \mathbb{N}^\star$, $\mathcal{R} \subseteq \mathcal{R}_\infty$, $\mathcal{X} \subset \mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
$\mathcal{R}_1 \leftarrow \mathcal{R}$, $\hat{\imath}_1 \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If there exists $r \in \mathcal{R}_t$ such that $r(X_{t+1}, X_{\hat{\imath}_t}) \geq 0$ {Decision rule}
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
$\mathcal{R}_t \leftarrow \{r \in \mathcal{R} : L_t(r) = 0\}$
$\hat{\imath}_t \in \arg\max_{i=1\ldots t} f(X_i)$

3. Output: Return $X_{\hat{\imath}_n}$

Figure 3.3: The RankOpt algorithm

that in active learning, one estimates a binary classifier $h : \mathcal{X} \to \{0,1\}$ where the goal in global optimization is to estimate the winner of a tournament deriving from the ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1,0,1\}$ and not the ranking rule itself.

Remark 3.12. (Adaptation to noisy evaluations) *It is noteworthy that the proposed optimization scheme could be extended to settings with noisy evaluations by slightly adapting the ideas developed in Dasgupta (2011) and Hanneke (2011). More precisely, a straightforward strategy would consist in using a relaxed version of the active subset $\mathcal{R}_{\delta,t} := \{r \in \mathcal{R} : L_t(r) \leq \min_{r \in \mathcal{R}} L_t(r) + UB_{\delta,t}\}$ where the term $UB_{\delta,t}$ comes out of some standard generalization bound on $\left|L_t(r_f) - \min_{r \in \mathcal{R}} L_t(r)\right|$ (see, e.g., Clémençon et al. (2010)).*

Remark 3.13. (Computational aspects) *Due to the theoretical nature and the genericity of the algorithm, several questions remain to be addressed in order to derive a practical implementation. In particular, the crucial steps of (i) identifying the set of ranking rules which minimize the empirical ranking loss and (ii) simulating the next evaluation points X_{t+1} with the rejection method might not be trivial in practice. Nevertheless, we point out that, under specific conditions, a complete implementation of the algorithm can be proposed (see Section 3.5 for further discussions on this topic).*

3.3.2 Convergence analysis

We state here some convergence properties of the RankOpt algorithm. The results are stated in a probabilistic framework. Recall however that the source of randomness comes from the random variables generated by the algorithm and not from the evaluations which are assumed to be noiseless. We start by casting an intermediate result that will be important in order to formulate the consistency property of the algorithm and the upper bound on the convergence rate.

Proposition 3.14. (Faster than Pure Random Search) *Let $\mathcal{X} \subset \mathbb{R}^d$ be any compact and convex set with non-empty interior, let $\mathcal{R}$ be any continuous ranking structure and let*

$f : \mathcal{X} \to \mathbb{R}$ be any function such that $r_f \in \mathcal{R}$. Then, for any $n \in \mathbb{N}^\star$, we have that $\forall y \in \mathbb{R}$,

$$\mathbb{P}\left(\max_{i=1\dots n} f(X_i) \geq y\right) \geq \mathbb{P}\left(\max_{i=1\dots n} f(X_i') \geq y\right)$$

where $X_1, \dots, X_n$ denotes a sequence of n evaluations points generated by RankOpt tuned with $\mathcal{R}$ over f and $X_1', \dots, X_n'$ is a sequence of n independent random variables uniformly distributed over $\mathcal{X}$.

One can then easily derive the next asymptotic result by combining Proposition 3.14 with the identifiability condition.

Corollary 3.15. (Consistency) *Consider the same assumptions as in Proposition 3.14. Then, under Condition 3.7, we have that*

$$\max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x \in \mathcal{X}} f(x).$$

Now we focus on the nonasymptotic performance of the algorithm. The next result provides our first finite-sample bound on the distance between the exact solution and the estimate provided by RankOpt.

Theorem 3.16. (Upper bound) *Suppose that the assumptions of Proposition 3.14 hold true. Then, under Condition 3.8, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\|x^\star - X_{\hat{\imath}_n}\|_2 \leq C_1 \cdot \left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$

where $C_1 = c_\alpha^{(2+\alpha)/(1+\alpha)} \operatorname{diam}(\mathcal{X})^{1/(1+\alpha)^2}$.

More surprisingly, a lower bound can also be derived by connecting the RankOpt algorithm to a theoretical algorithm defined below which uses the knowledge of the level sets of the unknown function.

Definition 3.17. (Pure Adaptive Search), *from Zabinsky and Smith (1992). We say that a sequence $X_1^\star, \dots, X_n^\star$ is distributed as a Pure Adaptive Search indexed by f over $\mathcal{X}$ if it follows the Markov process defined by:*

$$\begin{cases} X_1^\star \sim \mathcal{U}(\mathcal{X}) \\ X_{t+1}^\star | \, X_t^\star \sim \mathcal{U}(\mathcal{X}_t^\star) \qquad \forall t \in \{1 \dots n-1\} \end{cases}$$

where at each step $t \geq 1$ the next evaluation point $X_{t+1}^\star$ is sampled uniformly over the level set of the previous evaluation $\mathcal{X}_t^\star := \{x \in \mathcal{X} : f(x) \geq f(X_t^\star)\}$.

Precisely, the next result shows that the value of the highest evaluation observed by a Pure Adaptive Search is superior or equal, in the usual stochastic ordering sense, to the one observed by the RankOpt algorithm tuned with the same number of function evaluations.

Proposition 3.18. (Slower than Pure Adaptive Search) *Consider the same assumptions as in Proposition 3.14. Then, for any $n \in \mathbb{N}^\star$, we have that $\forall y \in \mathbb{R}$*

$$\mathbb{P}\left(\max_{i=1\dots n} f(X_i) \geq y\right) \leq \mathbb{P}\left(\max_{i=1\dots n} f(X_i^\star) \geq y\right)$$

where $X_1, \dots, X_n$ denotes a sequence of n evaluations points generated by RankOpt tuned with $\mathcal{R}$ over f and $X_1^\star, \dots, X_n^\star$ is a sequence of n evaluation points distributed as a Pure Adaptive Search indexed by f over $\mathcal{X}$.

As the performance of the algorithm can now be controlled by Proposition 3.18, it is then possible to establish a second finite-time bound on the distance between the exact solution and its approximation.

Theorem 3.19. (Lower bound) *Suppose that the assumptions of Proposition 3.14 hold true. Then, under Condition 3.8, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$C_2 \cdot e^{-\frac{(1+\alpha)^2}{d}\left(n+\sqrt{2n\ln(1/\delta)}+\ln(1/\delta)\right)} \leq \|x^\star - X_{\hat{\imath}_n}\|_2$$

where $C_2 = c_\alpha^{-(1+\alpha)(2+\alpha)} \operatorname{rad}(\mathcal{X})^{(1+\alpha)^2}$.

Note that, in addition to the following remarks, a complete discussion on the theoretical results obtained in this chapter can be found in the next section where an adaptive version of the algorithm is presented.

Remark 3.20. (Tightness of the bounds) *We stress that the RankOpt algorithm does achieve both the polynomial and exponential rates exhibited in Theorems 3.16 and 3.19 for specific ranking structures $\mathcal{R}$ and functions f. Indeed, noticing that the algorithm is equivalent to a Pure Random Search when $\mathcal{R}$ is set to $\mathcal{R}_\infty$, it can be easily shown by means of covering arguments that $\|X_{\hat{\imath}_n} - x^\star\|_2 = \Omega_{\mathbb{P}}(n^{-1/d})$ whenever f admits a unique maximum and $\mathcal{R} = \mathcal{R}_\infty$. Similarly, observing that the algorithm is equivalent to a pure adaptive search when $\mathcal{R}$ is set to $\{r_f\}$, it can also be shown that $\|X_{\hat{\imath}_n} - x^\star\|_2 = O_{\mathbb{P}}(e^{-n/d(1+\alpha+\epsilon)})$ for any $\epsilon > 0$ whenever $\mathcal{R} = \{r_f\}$ and f has regular level sets but no flat parts (i.e. $\mu(\{x : f(x) = y\}) = 0$ for all $y \in \operatorname{Im}(f)$). As these bounds actually match the one reported above when $\alpha = 0$, we thus deduce that the RankOpt algorithm does indeed achieve the near-optimal exponential and polynomial rates of $\Theta^*_{\mathbb{P}}(e^{-n/d})$ and $\Theta_{\mathbb{P}}(n^{-1/d})$ on any function f with $(1,0)$-regular level sets and no flat parts when the ranking structure $\mathcal{R}$ is respectively set to $\{r_f\}$ and $\mathcal{R}_\infty$.*

Remark 3.21. (Gap between the bounds) *As a consequence of the previous remark, we underline that, whereas the upper and lower bounds reported in Theorems 3.16 and 3.19 display display very different convergence rates, this gap can not be significantly reduced without imposing further conditions on both the function f and the ranking structure $\mathcal{R}$ set as input. Indeed, observe that since the algorithm can achieve both the rates of $\Theta^*_{\mathbb{P}}(e^{-n/d})$ and $\Theta_{\mathbb{P}}(n^{-1/d})$ on the same function f depending on choice of the ranking structure $\mathcal{R}$, then the gap between any lower and upper bounds on the convergence rate will necessarily be at least of the order of $[e^{-n/d}, n^{-1/d}]$ as long as no further assumptions are made on f and $\mathcal{R}$.*

Remark 3.22. (Choice of the ranking structure) *Finally, some simple indications on how to choose in practice the ranking structure $\mathcal{R}$ set as input can be deduced from the previous remarks. Recall indeed that since the algorithm achieves its best performance when $\mathcal{R}$ is set to $\{r_f\}$, then an ideal but realistic ranking structure $\mathcal{R}$ should be (i) large enough so it might contain r_f and (ii) as small as possible in order to obtain the exponentially decreasing rate reported when $\mathcal{R} = \{r_f\}$. Even though these indications serve opposite goals, we will see how to combine them carefully in the next section in order to derive an adaptive version of*

Input: $n \in \mathbb{N}^\star$, $p \in (0,1)$, $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$, $\mathcal{X} \subset \mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$
$\mathcal{R} \leftarrow \mathcal{R}_1$, $\hat{\imath}_1 \leftarrow 1$

2. Iterations: Repeat while $t < n$
Let $B_{t+1} \sim \mathcal{B}(p)$
If $B_{t+1} = 1$ {Exploration}
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $B_{t+1} = 0$ {Exploitation}
Let $X_{t+1} \sim \mathcal{U}(\{x \in \mathcal{X} : \exists r \in \mathcal{R} \text{ s.t. } r(x, X_{\hat{\imath}_t}) \geq 0\})$
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$
$\hat{\imath}_t \in \arg\max_{i=1\dots t} f(X_i)$
$\widehat{k}_t \leftarrow \min\{k \in \mathbb{N}^\star : \min_{r\in\mathcal{R}_k} L_t(r) = 0\}$ {Model Selection}
$\mathcal{R} \leftarrow \{r \in \mathcal{R}_{\widehat{k}_t} : L_t(r) = 0\}$

3. Output: Return $X_{\hat{\imath}_n}$

Figure 3.4: The AdaRankOpt algorithm

the algorithm that automatically selects the ranking structure $\mathcal{R}$ among a series of ranking structures of different complexities.

3.4 Adaptive algorithm and stopping time analysis

We consider here the problem of optimizing an unknown function f when no information is available on its induced ranking rule r_f.

3.4.1 The AdaRankOpt algorithm

The AdaRankOpt algorithm (shown in Figure 3.4) is an extension of the RankOpt algorithm which involves model selection. We consider a parameter $p \in (0,1)$ and a nested sequence of ranking structures $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ satisfying

$$\mathcal{R}_1 \subset \mathcal{R}_2 \subset \cdots \subset \mathcal{R}_\infty. \tag{3.1}$$

The algorithm is initialized by evaluating the function at a point X_1 uniformly distributed over $\mathcal{X}$ and by considering the smallest ranking structure $\mathcal{R}_1$ of the sequence. At each iteration $t < n$, a Bernoulli random variable B_{t+1} of parameter p is sampled. If $B_{t+1} = 1$, the algorithm explores the space by evaluating the function at a point uniformly sampled over $\mathcal{X}$. If $B_{t+1} = 0$, the algorithm exploits the previous evaluations by making an iteration of the RankOpt algorithm with the smallest ranking structure $\mathcal{R}_{\hat{k}_t}$ of the sequence that probably contains the true ranking r_f. Once a new evaluation $f(X_{t+1})$ has been made, the index $\widehat{k}_t := \min\{k \in \mathbb{N}^\star : \min_{r\in\mathcal{R}_k} L_t(r) = 0\}$ of the smallest ranking structure of the sequence $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ which contains a ranking rule consistent with the sample is up-

dated. Hence, the parameter p drives the trade-off between the exploitation phase and the exploration phase which prevents the algorithm from getting stuck in a local maximum.

3.4.2 Theoretical properties of AdaRankOpt

We start by casting the consistency result for the algorithm.

Proposition 3.23. (Consistency) *Fix any $p \in (0,1)$ and let $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ be any nested sequence of ranking structures. Then, under Condition 3.7, we have that*

$$\max_{i=1\dots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x).$$

where $X_1,\dots,X_n$ denotes a sequence of n evaluation points generated by AdaRankOpt tuned with p and $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ over f.

The previous result shows that the adaptive version of the algorithm remains consistent over the same set of identifiable functions regardless of its tuning (*e.g.* the choice of the sequence of ranking structures and the value of p). We thus have to examine its nonasymptotic performance in order to understand the impact of these parameters on its behavior.

We begin the finite-time analysis by investigating the number of iterations required to identify a ranking structure which contains the ranking rule r_f induced by the unknown function.

Definition 3.24. (Stopping Time) *Let $k^\star = \min\{k \in \mathbb{N}^\star : r_f \in \mathcal{R}_k\}$ be the index of the smallest ranking structure of the sequence which contains r_f and let $\{\widehat{k}_t\}_{t\in\mathbb{N}^\star}$ be the sequence of random variables driving the model selection defined in the algorithm. We define the stopping time which corresponds to the number of iterations required to identify the index $k^\star$ as*

$$\tau_{k^\star} := \min\{t \in \mathbb{N}^\star : \widehat{k}_t = k^\star\}.$$

In order to bound $\tau_{k^\star}$, we need to control the complexity of the sequence of ranking structures $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$. Let us denote by $L(r) = \mathbb{P}(r(X,X') \neq r_f(X,X'))$ the true ranking loss where (X,X') is a couple of independent random variables uniformly distributed over $\mathcal{X}$ and define the Rademacher average of a ranking structure $\mathcal{R}$ given r_f as

$$R_n(\mathcal{R}) := \sup_{r\in\mathcal{R}} \frac{1}{\lfloor n/2\rfloor} \left| \sum_{i=1}^{\lfloor n/2\rfloor} \epsilon_i \cdot \mathbb{I}\big\{r(X_i, X_{\lfloor n/2\rfloor+i}) \neq r_f(X_i, X_{\lfloor n/2\rfloor+i})\big\} \right|$$

where $\{X_i\}_{i=1}^n$ are n independent copies of $X \sim \mathcal{U}(\mathcal{X})$ and $\{\epsilon_i\}_{i=1}^{\lfloor n/2\rfloor}$ are $\lfloor n/2\rfloor$ independent Rademacher random variables (*i.e.*, random symmetric sign variables), also independent of $\{X_i\}_{i=1}^n$.

Proposition 3.25. (Stopping Time Upper Bound) *Assume that the index $k^\star > 1$ is finite, assume that $\inf_{r\in\mathcal{R}_{k^\star-1}} L(r) > 0$ and assume that there exists a constant $K > 0$ such that $\forall n \in \mathbb{N}^\star$, the Rademacher complexity of $\mathcal{R}_{k^\star-1}$ satisfies $\mathbb{E}\left[R_n(\mathcal{R}_{k^\star-1})\right] \leq \sqrt{K/n}$. Then, for any $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\tau_{k^\star} \leq \frac{10}{p} \cdot \left(\frac{K + \ln(2/\delta)}{\inf_{r\in\mathcal{R}_{k^\star-1}} L(r)^2} \right).$$

In the situation described above, the smallest ranking structure of sequence which contains the true ranking rule can be identified in a finite number of iterations. One can then recover an upper bound similar to the one of Theorem 3.16 where a ranking structure containing r_f is assumed to be known.

Theorem 3.26. (Upper Bound) *Suppose that the assumptions of Proposition 3.25 hold ture. Then, under Condition 3.8, for any $\delta \in (0,1)$ and $n \in \mathbb{N}^\star$, we have with probability at least $1-\delta$,*

$$\|X_{\hat{\imath}_n} - x^\star\|_2 \leq C_1 \cdot \left(\frac{11(K+\ln(4/\delta))}{p \inf_{r \in \mathcal{R}_{k^\star-1}} L(r)^2}\right) \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}$$

where C_1 is the same constant as in Theorem 3.16 and $X_{\hat{\imath}_n}$ denotes the output of AdaRankOpt$(n, f, \mathcal{X}, p, \{\mathcal{R}_k\}_{k \in \mathbb{N}^\star})$.

The following remarks provide some insights on the different conditions and quantities involved in the theorem.

Remark 3.27. (On the complexity assumption) *As pointed out in Clémençon (2011) (see Remark 2 therein), standard VC-type arguments can be used in order to bound $\mathbb{E}[R_n(\mathcal{R}_{k^\star-1})]$. More specifically, if the set of functions $\mathcal{F}_{k^\star-1} = \{(x,x') \in \mathcal{X}^2 \mapsto \mathbb{I}\{r(x,x') \neq r_f(x,x')\} \mid r \in \mathcal{R}_{k^\star-1}\}$ is a VC major class with finite VC dimension V, then $\mathbb{E}\left[R_n(\mathcal{R}_{k^\star-1}))\right] \leq c \cdot \sqrt{V/n}$ for a universal constant $c > 0$. This covers, in particular, the classes of polynomial and sinusoidal ranking rules of any degree $k^\star \geq 1$.*

Remark 3.28. (On the infimum ranking loss) *In order to grasp the meaning of the term $\inf_{r \in \mathcal{R}_{k^\star-1}} L(r)$, observe that since the function $\rho : (r,r') \mapsto \mathbb{P}_{X,X' \sim \mathcal{U}(\mathcal{X})}(r(X,X') \neq r'(X,X'))$ defines a metric over the product space $\mathcal{R}_\infty \times \mathcal{R}_\infty$, then the infimum ranking loss $\inf_{r \in \mathcal{R}_{k^\star-1}} L(r) = \inf_{r \in \mathcal{R}_{k^\star-1}} \rho(r, r_f)$ can be interpreted as a measure of the distance between the ranking rule r_f and the ranking structure $\mathcal{R}_{k^\star-1}$. As a consequence of this observation, we point out that the condition $\inf_{r \in \mathcal{R}_{k^\star-1}} L(r) > 0$ can be easily checked to be fulfilled for the sequences of polynomial and sinusoidal ranking rules whenever $r_f \in \mathcal{R}_{k^\star}$ for some $k^\star > 1$ by combining their parametric representation with the definition of the metric $\rho(\cdot,\cdot)$.*

3.4.3 Comparison with previous works

Our interest here is to compare the theoretical results obtained in this chapter to existing results in the global optimization literature. We consider three types of algorithms.

DIRECT and SOO (Jones et al. (1993) and Munos (2014)). These algorithms use a splitting technique of the search space and sequentially evaluate the function on subdivisions of the space that have recorded the highest evaluation among all the subdivisions of similar size. To the best of our knowledge, there is no finite-time analysis of the DIRECT algorithm (only the consistency was proven by Finkel and Kelley (2004)). However, Munos (2014) identified some local smoothness conditions allowing to derive a finite-time analysis of the algorithms. Precisely, assuming there exists $x^\star \in \mathcal{X}$, $\eta, c_1, c_2, \nu > 0$ and $\alpha \geq 0$ such that $\forall x \in B(x^\star, \eta)$, $c_1 \|x^\star - x\|^\nu \leq f(x^\star) - f(x) \leq c_2 \|x^\star - x\|^{\nu/(1+\alpha)}$ for some norm $\|\cdot\|$ (*e.g.*, ℓ_2, ℓ_∞), the author reports for the SOO algorithm a polynomial upper bound on the difference between the maximum and its estimation of $\max_{x \in \mathcal{X}} f(x) - f(X_{\hat{\imath}_n}) = O(n^{-\nu/\alpha d})$ when $\alpha > 0$ and an exponential decay of

$O(e^{-c\nu\sqrt{n}})$ for some $c > 0$ when $\alpha = 0$. As a comparison, we obtain for AdaRankOpt a polynomial bound of $\max_{x\in\mathcal{X}} f(x) - f(X_{\hat{\imath}_n}) = O_{\mathbb{P}}(n^{-\nu/d})$ for all $\alpha \geq 0$ by assuming that both the conditions of Proposition 3.25 and the local smoothness condition are fulfilled. Hence the bound we obtain turns out to be better when $\alpha > 1$ and much worse for $\alpha < 1$. This is consistent with the fact that while the asymmetry in the smoothness of the function around its maximum (captured here by α) strongly impacts the performance of SOO in both ways, it does not impact AdaRankOpt which remains invariant to the variations of the local smoothness of the unknown function around its maximum.

Evolution Strategies (Rechenberg (1971)). We now consider the class of (μ,λ)-Evolution Strategies which use mutation, recombination, and selection in order to iteratively evolve the set of evaluation points. As far as we know, no consistency results or generic upper bounds have been proven for these algorithms. However, Teytaud and Fournier (2008) were able to derive exponential lower bounds for several extensions of the (μ,λ)-ES using the VC-dimension V of the level sets of the unknown function. Precisely, they showed that if V is finite, then $\|X_{\hat{\imath}_n} - x^\star\|_2 = \Omega_{\mathbb{P}}(e^{-c(V)n/d})$ where $c(V)$ is a constant that depends on both the extension under consideration and V. Moreover Auger (2005) also analyzed the convergence of the $(1,\lambda)$-SA-ES algorithm on the simple sphere function $f(x) = -\|x\|_2^2$ and proved specific conditions on the parameters of the algorithm in order to ensure that $\ln(\|X_{\hat{\imath}_n}\|_2)/n \xrightarrow{a.s.} c$ for some constant $c \in \mathbb{R}$. However, since the sign of the limit of $\ln(\|X_{\hat{\imath}_n}\|_2)/n$ remains unknown, this result only proves the exponential convergence or divergence of the algorithm. Moreover, we point out the results reported in those works can not be directly compared to the one obtained in this chapter, as they are of opposite nature. Recall indeed that while we were able to derive (i) a generic upper bound for AdaRankOpt and (ii) a lower bound for its nonadaptive version, they obtained on the contrary (i) generic lower bounds for various extensions of the (μ,λ)-ES and (ii) an asymptotic upper bound which might be only valid for a specific version of the algorithm in the case where f is the sphere function.

Expected Improvement Strategy (Močkus (1975)). The last algorithm we consider is a Bayesian optimization strategy which selects at each step $t \geq 2$ an evaluation point $x_{t+1} \in \arg\max_{x\in\mathcal{X}} \mathbb{E}_{f\sim\pi}[\max(f(x) - \max_{i=1\ldots t} f(x_i), 0) | \{(x_i, f(x_i))\}_{i=1}^t]$ where f is assumed to be drawn from a law π set as input. Vazquez and Bect (2010) showed that when π is a fixed Gaussian process prior with a finite smoothness, the EI strategy converges on the maximum of any function f of the reproducing kernel Hilbert space $\mathcal{H}$ canonically attached to π. Moreover Bull (2011) went on to prove that an adaptive version of the EI algorithm they define could achieve a near-optimal polynomial bound of $\max_{x\in\mathcal{X}} f(x) - f(X_{\hat{\imath}_n}) = O_{\mathbb{P}}(n^{-\nu/d})$ for all $f \in \mathcal{H}$ when π is a prior of smoothness $\nu \geq 0$. As a comparison, considering that both the conditions of Proposition 3.25 are fulfilled and that $f \in \mathcal{H}$, we obtain for AdaRankOpt the exact same polynomial bound of $\max_{x\in\mathcal{X}} f(x) - f(X_{\hat{\imath}_n}) = O_{\mathbb{P}}(n^{-\nu/d})$. However, this similarity simply comes from the fact that the author also used a very similar—and potentially suboptimal—covering argument of the search space in order to prove their result.

These comparisons suggest that although the upper bounds provided in this chapter could certainly be improved in order to obtain the exponentially decreasing loss exhibited in Theorem 3.19 and observed in Munos (2014). Nonetheless, as detailed in Remark 3.21, such

an analysis would require a refinement of the characterization of a real-valued function with regards to a ranking structure and is therefore left as future work.

3.5 Implementation and computational aspects

In this section, we discuss some technical aspects involved in the practical implementation of AdaRankOpt. In particular, we provide some equivalences that can be used in order to implement the algorithm for the classes of ranking structures introduced in Section 3.2 without explicitly maintaining the active subset of consistent ranking rules.

3.5.1 Notations

We collect here the specific notations used in this section. For any sample $\{(X_i, f(X_i))\}_{i=1}^{t+1}$ of $t+1$ function evaluations with distinct values (*i.e.* such that $f(X_i) \neq f(X_j)$ for all $i \neq j$), we denote by $(1), (2), \ldots, (t+1)$ the indexes corresponding to the strictly increasing reordering: $f(X_{(1)}) < f(X_{(2)}) < \cdots < f(X_{(t+1)})$. For any dimension $d \geq 1$, we respectively denote by $\vec{0} = (0, \ldots, 0) \in \mathbb{R}^d$ and by $\vec{1} = (1, \ldots, 1) \in \mathbb{R}^d$ the zero and the unit vector of $\mathbb{R}^d$. The notation $x \succeq x'$ corresponds to the component-wise inequality (*i.e.* $x \succeq x' \in \mathbb{R}^d \;\Leftrightarrow\; \forall i \in \{1 \ldots d\},\ x_i \geq x'_i$) and we denote by $\mathrm{ConvHull}\{x_i\}_{i=1}^{t}$ the convex hull of any set $\{x_i\}_{i=1}^{t}$ of $t \geq 1$ points in $\mathbb{R}^d$. For any degree $k \geq 1$, the function that maps $\mathbb{R}^d$ into the corresponding polynomial feature space of degree k is denoted by $\Phi_k : \mathbb{R}^d \to \mathbb{R}^{\dim(\Phi_k)}$ where $\dim(\Phi_k) = \binom{k+d}{d} - 1$. For instance, in the case where $k = d = 2$, we have that $\Phi_2(x) = (x_1, x_2, x_1 x_2, x_1^2, x_2^2) \in \mathbb{R}^5$ for all $x = (x_1, x_2) \in \mathbb{R}^2$. Finally, we denote by $\mathrm{M}_t^{\Phi_k} = [C_1 \mid \cdots \mid C_t]$ the $(\dim(\Phi_k), t)$-matrix with its i-th column C_i equal to $(\Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}))^\mathsf{T}$ and we denote for all $i \leq t+1$ by $\mathrm{M}_i = [C_1 \mid \cdots \mid C_i]$ the (d, i)-matrix where its j-th column C_j is equal to $X_{(t+2-j)}^\mathsf{T}$.

3.5.2 General ranking structures

Suppose now that we have collected a sample $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ of $t \geq 2$ observations generated by AdaRankOpt tuned with any nested sequence of ranking structures $\{\mathcal{R}_k\}_{k \in \mathbb{N}^*}$. We address here the questions of **(i)** sampling the next evaluation point X_{t+1} and **(ii)** updating the index $\widehat{k}_{t+1}$ of the model selection once $f(X_{t+1})$ has been evaluated.

(i) First, we consider the problem of sampling the next evaluation point $X_{t+1} \sim \mathcal{U}(\mathcal{X}_t)$ over the non-trivial subset $\mathcal{X}_t := \{x \in \mathcal{X} : \exists r \in \mathcal{R}_{\widehat{k}_t}$ such that $L_t(r) = 0$ and $r(x, X_{\hat{\imath}_t}) \geq 0\}$. To do so, we propose to use the rejection which consists in sampling $X' \sim \mathcal{U}(\mathcal{X})$ until $X' \in \mathcal{X}_t$. We thus need to set up a procedure that tests if any point $X' \in \mathcal{X}$ belongs to $\mathcal{X}_t$. By definition of $\mathcal{X}_t$, we know that $X' \in \mathcal{X}_t$ if and only if there exists a ranking rule r in $\mathcal{R}_{\widehat{k}_t}$ which satisfies $L_t(r) = 0$ and $r(X', X_{\hat{\imath}_t}) = 0$ *or* 1. Therefore, we obtain by rewriting the previous statement in terms of minimal error that $X' \in \mathcal{X}_t$ if and only if:

- either $\min_{r \in \mathcal{R}_{\widehat{k}_t}} L_{t+1}(r) = 0$ where the empirical ranking loss is taken over the sample $\{(X_i, f(X_i))\}_{i=1}^{t} \cup (X', f(X_{\hat{\imath}_t}))$ (case $r(X', X_{\hat{\imath}_t}) = 0$);

- or $\min_{r\in\mathcal{R}_{\widehat{k}_t}} L_{t+1}(r) = 0$ where $L_{t+1}(\cdot)$ is taken over the sample $\{(X_i, f(X_i))\}_{i=1}^{t} \cup (X', f(X_{\hat{\imath}_t}) + c)$ for any positive constant $c > 0$ (case $r(X', X_{\hat{\imath}_t}) = 1$).

Hence, X_{t+1} can be generated by sequentially sampling $X' \sim \mathcal{U}(\mathcal{X})$ until there exists a ranking rule r in $\mathcal{R}_{\widehat{k}_t}$ that perfectly ranks the initial set of t observations where we added a supplementary ghost evaluation $\{\{(X_i, f(X_i))\}_{i=1}^{t} \cup (X', f(X_{\hat{\imath}_n}) + c)\}$ for some $c \geq 0$.

(ii) We now consider the problem of updating the index $\widehat{k}_{t+1}$ of the model selection once $f(X_{t+1})$ has been evaluated. Since $\{\mathcal{R}_k\}_{k\in\mathbb{N}^\star}$ forms, by assumption, a nested sequence, it necessarily follows that the sequence of indexes $\{\widehat{k}_t\}_{t\in\mathbb{N}^\star}$ is also increasing. One can thus write that $\widehat{k}_{t+1} = \widehat{k}_t + \min\{i \in \mathbb{N}^\star : \min_{r\in\mathcal{R}_{\widehat{k}_{t+i}}} L_{t+1}(r) = 0\}$ where the empirical ranking loss $L_{t+1}(\cdot)$ is computed over the sample $\{(X_i, f(X_i))\}_{i=1}^{t+1}$. Therefore, the index $\widehat{k}_{t+1}$ can be updated by sequentially testing if $\min_{r\in\mathcal{R}_{\widehat{k}_{t+i}}} L_{t+1}(r) = 0$ for $i = 0, 1, 2, \ldots$

As shown above, both the steps **(i)** and **(ii)** can be done using a single generic procedure that determines if $\min_{r\in\mathcal{R}_k} L_{t+1}(r) = 0$ holds true for any ranking structure $\mathcal{R}_k$ of the sequence with $k \geq 1$ and where the empirical ranking loss $L_{t+1}(\cdot)$ is computed over any sample of $t+1$ function evaluations. In the next subsections, we provide some equivalences that can be used in order to design such a procedure for the classes of ranking structures introduced in Section 3.2. For simplicity, we will consider in the sequel that all the function evaluations of the sample have distinct values.

3.5.3 Polynomial and sinusoidal ranking rules

We consider here the sequence of polynomial ranking rules $\{\mathcal{R}_{\mathcal{P}_k}\}_{k\in\mathbb{N}^\star}$ and we recall that $\Phi_k(\cdot)$ denotes the function that maps $\mathbb{R}^d$ into the corresponding polynomial feature space of degree $k \geq 1$. However, we point out that the results stated below can easily be adapted for the sequence of sinusoidal ranking rules by considering the adequate feature space. The first result we establish relates the existence of a consistent polynomial ranking rule to the linear separability of a sample-dependent set of points which belong to the corresponding feature space.

Proposition 3.29. (Separability) *Let $(X_1, f(X_1)), \ldots, (X_{t+1}, f(X_{t+1}))$ be any sample of $t+1$ function evaluations with distinct values. Then, there exists a polynomial ranking rule of degree $k \geq 1$ that perfectly ranks the sample (i.e. $\min_{r\in\mathcal{R}_{\mathcal{P}_k}} L_{t+1}(r) = 0$) if and only if there exists an axis $\omega \in \mathbb{R}^{\dim(\Phi_k)}$ satisfying:*

$$\langle \omega, \Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}) \rangle > 0, \ \forall i \in \{1 \ldots t\}.$$

where $(1), (2), \ldots (t+1)$ denote the indexes of the strictly increasing reordering of the sample.

Unfortunately, the equivalence exhibited in Proposition 3.29 might not be always convenient in practice since the computational cost of estimating such an axis $\omega \in \mathbb{R}^{\dim(\Phi_k)}$ can be prohibitive in the case of feature spaces with large dimensionalities $\dim(\Phi_k)$. Nonetheless, as the previous result only makes the link with the *existence* of a separating axis, one can then use the following lemma presented in the generic framework of binary classification and illustrated in Figure 3.5 in order get an equivalence generally easier to check in practice.

Lemma 3.30. *Let $(x_1,y_1),\dots,(x_t,y_t)$ be any set of binary classification samples where $(x_i,y_i) \in \mathbb{R}^d \times \{-1,+1\}$. Then, there exists a separating axis $\omega \in \mathbb{R}^d$ satisfying*

$$y_i \cdot \langle \omega, x_i \rangle > 0,\ \forall i \in \{1 \dots t\}$$

if and only if

$$\vec{0} \notin \textsc{ConvHull}\{y_i \cdot x_i\}_{i=1}^t.$$

One can then deduce from the combination of Proposition 3.29 and Lemma 3.30 that testing the existence of a consistent polynomial ranking rule can simply be performed by checking the emptiness of a specific polyhedron built from the sample as detailed in the next corollary.

Corollary 3.31. *Consider the same assumptions as in Proposition 3.29. Then, there exists a polynomial ranking rule of degree $k \geq 1$ that perfectly ranks the sample if and only if the polyhedral set $\Omega_t^{\Phi_k}$ defined by*

$$\Omega_t^{\Phi_k} := \left\{ \lambda \in \mathbb{R}^t :\ \mathrm{M}_t^{\Phi_k} \lambda^\mathsf{T} = \vec{0},\ \langle \vec{1}, \lambda \rangle = 1,\ \lambda \succeq \vec{0} \right\}$$

is empty where $\mathrm{M}_t^{\Phi_k} = [C_1 \mid \cdots \mid C_t]$ is the $(\dim(\Phi_k), t)$-matrix with its i-th column C_i equal to $(\Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}))^\mathsf{T}$.

As a full implementation of the algorithm can be derived at this point (see Figure 3.6 below), a few comments are in order.

Remark 3.32. (Algorithmic aspects) *Notice that, in practice, the problem of testing the emptiness of a polyhedral set admits a tractable solution. Indeed, it can be seen as the problem of determining if a particular linear program admits a feasible point and can therefore be solved with the simplex algorithm. For further details on this topic, we refer to Chapter 11.4 in Boyd and Vandenberghe (2004) where practical examples as well as algorithmic solutions are discussed.*

Remark 3.33. (Numerical complexity) *In contrast, the numerical complexity of the proposed implementation can not be tracked precisely due to the stochastic nature of the rejection method. Nevertheless, we point out that a simple union bound indicates that the complexity of generating the next evaluation point $X_{t+1} | \{X_i\}_{i=1}^t \sim \mathcal{U}(\mathcal{X}_t)$ given a sample $\{(X_i, f(X_i))\}_{i=1}^t$ is upper bounded, with probability at least $1-\delta$, by the complexity of testing the emptiness of a polyhedron multiplied by $\lceil \ln(\delta) / \ln(1 - \mu(\mathcal{X}_t)/\mu(\mathcal{X})) \rceil$. However, the value of the ratio $\mu(\mathcal{X}_t)/\mu(\mathcal{X})$ which controls the upper bound depends on both the random evaluations previously made and the nested structure of the level sets of the unknown function and can therefore not be developed further.*

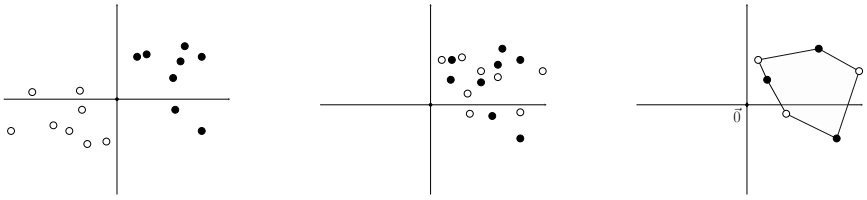


Figure 3.5: Illustration of Lemma 3.30. *Left:* A separable sample $\{(x_i,y_i)\}_{i=1}^n$. *Middle:* The sample $\{y_i \cdot x_i\}_{i=1}^n$. *Right:* The convex hull of $\{y_i \cdot x_i\}_{i=1}^n$ next to the zero vector.

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$, $\hat{k}_1 \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Let $B_{t+1} \sim \mathcal{B}(1/10)$
If $B_{t+1} = 1$
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
If $B_{t+1} = 0$
Bool $\leftarrow$ True
While *Bool* {Rejection Method}
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
Let $\Omega_t^{\Phi_{\hat{k}_t}}$ be the polyhedron defined in Corollary 3.31 computed over the sample $\{(X_i, f(X_i))\}_{i=1}^t \cup (X_{t+1}, \max_{i=1\ldots t} f(X_i) + c)$ where $c > 0$ is any strictly positive constant and the degree k set to $\hat{k}_t$
Test if $\Omega_t^{\Phi_{\hat{k}_t}}$ is empty with the simplex algorithm
If $\Omega_t^{\Phi_{\hat{k}_t}}$ is empty
Bool $\leftarrow$ False

Evaluate $f(X_{t+1})$, $t \leftarrow t + 1$

$\hat{k}_t \leftarrow \hat{k}_{t-1}$, *Bool* $\leftarrow$ True
While *Bool* {Model Selection}
Let $\Omega_t^{\Phi_{\hat{k}_t}}$ be the polyhedron defined in Corollary 3.31 computed over the sample $\{(X_i, f(X_i))\}_{i=1}^t$ with a degree k set to $\hat{k}_t$
Test if $\Omega_t^{\Phi_{\hat{k}_t}}$ is empty with the simplex algorithm
If $\Omega_t^{\Phi_{\hat{k}_t}}$ is empty
Bool $\leftarrow$ False
Else
$\hat{k}_t \leftarrow \hat{k}_t + 1$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1\ldots n} f(X_i)$

Figure 3.6: Implementation of the AdaRankOpt algorithm with the sequence of polynomial ranking structures and with a parameter p set to 1/10.

3.5.4 Convex ranking rules

We now consider the nonparametric sequence of convex ranking rules $\{\mathcal{R}_{\mathcal{C}_k}\}_{k \in \mathbb{N}^\star}$. The equivalences provided below essentially rely on the fact that any bipartite ranking rule can be approximated by overlaying a finite sequence binary classifiers as previously shown in the work of Clémençon and Vayatis (2010). We start with the one-dimensional case.

Proposition 3.34. (Overlaying classifiers) *Set $d = 1$ and assume that we have collected a sample $\{(X_i, f(X_i))\}_{i=1}^{t+1}$ of $t + 1$ function evaluations with distinct values. Then, there exists a convex ranking rule of degree $k \geq 1$ that perfectly ranks the sample if and only if*

there a exists a sequence of classifiers $\{h_i\}_{i=1}^{t+1}$ *of the form* $h_i(x) = \sum_{m=1}^{k} \mathbb{I}\{l_{i,m} \leq x \leq u_{i,m}\}$ *satisfying:*

1. $h_i(X_{(j)}) = \mathbb{I}\{j \geq i\}, \forall (i,j) \in \{1 \dots t+1\}^2$;
2. $h_1 \geq h_2 \geq \cdots \geq h_{t+1}$.

where $(1),(2)\dots(t+1)$ *denote the indexes of the strictly increasing reordering of the sample.*

In the specific case where $d > 1$ and $k = 1$, we further argue that checking the existence of a consistent and finite collection of nested convex classifiers can be performed by determining the emptiness of a cascade of polyhedral sets.

Proposition 3.35. *Set any* $d \in \mathbb{N}^\star$ *and assume that we have collected a sample* $\{(X_i, f(X_i))\}_{i=1}^{t+1}$ *of* $t+1$ *function evaluations with distinct values. Then, there exists a convex ranking rule of degree* $k = 1$ *that perfectly ranks the sample if and only if for each* $i = 1, \dots, t$*, the polyhedral set* Ω_i *defined by*

$$\Omega_i := \left\{ \lambda \in \mathbb{R}^i : \ \mathrm{M}_i \lambda = X_{(t+1-i)}^\mathsf{T}, \ \langle \vec{1}, \lambda \rangle = 1, \ \lambda \succeq \vec{0} \right\}$$

is empty where $\mathrm{M}_i = [C_1 \mid \cdots \mid C_i]$ *is the* (d,i)*-matrix with its* j*-th column* C_j *is equal to* $X_{(t+2-j)}^\mathsf{T}$.

As an example, an implementation of the RankOpt algorithm can be found in Figure 3.7.

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $f(X_1)$, $t \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Let $X_{t+1} \sim \mathcal{U}(\mathcal{X})$
Bool $\leftarrow$ *True*
For $i = 1 \dots t$
Let Ω_i be the polyhedron defined in Proposition 3.35 computed with the sample $\{(X_i, f(X_i))\}_{i=1}^{t} \cup (X_{t+1}, \max_{i=1\dots t} f(X_i) + c)$ with any $c > 0$
Use the simplex algorithm to test if Ω_i is empty
If Ω_i is not empty
Bool $\leftarrow$ *False*
If *Bool* $=$ *True*
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$

3. Output: Return $X_{\hat{\imath}_n}$

Figure 3.7: Implementation of the RankOpt algorithm with the convex ranking structure of degree $k = 1$.

3.6 Discussion and perspectives

The major contribution of this work was to show how to apply the principles of bipartite ranking to the global optimization problem. We introduced two novel global optimization strategies based on a sequential estimation of the ranking rule induced by the unknown function: RankOpt which requires the knowledge of a ranking structure containing the induced ranking rule of the unknown function and its adaptive version AdaRankOpt which performs model selection. A theoretical analysis of the algorithms is provided and empirical results based on synthetic and real problems are provided in Chapter 4, demonstrating the competitiveness of the adaptive version of the algorithm with regards to the existing state-of-the-art global optimization methods. Possible future research directions include:

- Deriving an implementation of the algorithm for the classes of convex ranking rules of degree $k > 2$ when $d > 1$.
- Refining the characterization of real-valued functions with regards to a ranking structure in order to identify the classes of functions providing the exponentially decreasing loss obtained in the lower bound.
- Determining better values for the exploration/exploitation parameter p as well as for the choice of the sequence of ranking structures set as input. A promising approach would consist in using dynamic values for p_t depending on the previous evaluations and considering at the same time multiple sequences of ranking structures (*e.g.,* plonymial and convex ranking structures) and picking among them the one that explains the function evaluations with the ranking rule of lowest degree.
- Extending the algorithms to settings with noisy evaluations and investigating alternative sampling strategies. For instance, a Bayesian approach would consist in setting a distribution over the ranking rule underlying the unknown function to sample the next evaluation points. As an example, since the class of polynomial ranking rules of degree $k \geq 1$ can be parametrized as follows:

$$\mathcal{R}_{\mathcal{P}_k} = \{(x,x') \in \mathcal{X}^2 \mapsto \langle \omega, \Phi_k(x) - \Phi_k(x') \rangle | \omega \in \mathcal{S}^{\dim(\Phi_k)}\}$$

  where $\mathcal{S}^{\dim(\Phi_k)}$ denotes the $\dim(\Phi_k)$-dimensional hypersphere of unit radius, one can easily set a distribution on the unknown ranking rule (*e.g.,* $r_f(x,x') = \langle \omega, \Phi_k(x) - \Phi_k(x') \rangle$ where $\omega \sim \mathcal{U}(\mathcal{S}^{\dim(\Phi_k)})$). Given a distribution on r_f, it is then possible to define various strategies aiming at minimizing the expected value of the size of the set of potential maximizers.

3.7 Proofs

In this section, we provide all the proofs of the results presented in this chapter.

3.7.1 Proofs of Section 3.2 and preliminary results

We develop here the proof for the equivalence class of real-valued functions sharing the same induced ranking stated in Proposition 3.2.

Proof of proposition 3.2 ($\Leftarrow$) Assume that there exists a strictly increasing function $\psi : \mathbb{R} \to \mathbb{R}$ such that $h = \psi \circ f$. Since ψ is strictly increasing, it directly follows that $\forall (x,x') \in \mathcal{X}^2$,

$$r_h(x,x') = \text{sgn}(\psi \circ f(x) - \psi \circ f(x')) = \text{sgn}(f(x) - f(x')) = r_f(x,x').$$

($\Rightarrow$) Assume now that $\forall (x,x') \in \mathcal{X}^2, r_f(x,x') = r_h(x,x')$. Observe first that if $\forall (x,x') \in \mathcal{X}^2, r_f(x,x') = r_h(x,x') = 0$, then both $f = c_1$ and $h = c_2$ are constant over $\mathcal{X}$ and we have that $h = \psi \circ f$ where $\psi : x \mapsto x + (c_2 - c_1)$ is a strictly increasing function. We now consider the case where f is not constant over $\mathcal{X}$ and we show that there exists a strictly increasing function $\psi : \mathbb{R} \to \mathbb{R}$ such that $f = \psi \circ M$ where $M : \mathcal{X} \to [0,1]$ is defined for all $x \in \mathcal{X}$ by:

$$M : x \mapsto \int_{x' \in \mathcal{X}} \mathbb{I}\{r_f(x,x') < 0\} \, \mathrm{d}x' = \mu(\{x' \in \mathcal{X} : f(x') < f(x)\}).$$

To properly define ψ, we first need to ensure that the function f is constant over the iso-level set $M^{-1}(y) = \{x \in \mathcal{X} : M(x) = y\}$ for all $y \in \text{Im}(M)$. To do so, fix any $y \in \text{Im}(M)$, pick any $(x_1,x_2) \in M^{-1}(y) \times M^{-1}(y)$ and assume by contradiction and without loss of generality that $f(x_1) < f(x_2)$. As the ranking rules r_f and r_h are assumed to be equal over $\mathcal{X} \times \mathcal{X}$, we have (i) $h(x_1) < h(x_2)$ and (ii) $M(x_i) = \mu(\{x' : h(x') < h(x_i)\})$ for $i \in \{1,2\}$. However, putting (i) and (ii) altogether with the continuity of h leads us to the contradiction:

$$M(x_1) = \mu(\{x' \in \mathcal{X} : h(x') < h(x_1)\}) < \mu(\{x' \in \mathcal{X} : h(x') < h(x_2)\}) = M(x_2)$$

and we deduce that f is constant over any iso-level set of M. Now, denoting by $f(M^{-1}(y))$ the unique value of f over $M^{-1}(y)$, we are ready to introduce the restriction of the function ψ over $\text{Im}(M)$ defined by:

$$\psi_{\text{Im}(M)} : y \in \text{Im}(M) \mapsto f(M^{-1}(y)) \in \mathbb{R}.$$

Since $\psi_{\text{Im}(M)}(M(x)) = f(x)$ for all $x \in \mathcal{X}$, we know from the continuity of h that $\psi_{\text{Im}(M)}(y_1) < \psi_{\text{Im}(M)}(y_2)$ for all $y_1 < y_2 \in \text{Im}(M) \times \text{Im}(M)$. Hence, $\psi_{\text{Im}(M)}$ is strictly increasing over $\text{Im}(f)$ and one can then write that $f = \psi \circ M$ where $\psi : \mathbb{R} \to \mathbb{R}$ is any strictly increasing extension of the function $\psi_{\text{Im}(M)}$ over $\mathbb{R}$. By reproducing the same steps as previously with h, it can also be shown that there also exists a strictly increasing function $\psi' : \mathbb{R} \to \mathbb{R}$ such that $h = \psi' \circ M$. Hence, one can write that $h = \psi' \circ M = (\psi' \circ \psi^{-1}) \circ f$ where $\psi' \circ \psi^{-1} : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function, which proves the result. □

The next lemma will be useful in order to control the volume of level sets of a function with (c_α, α)-regular level sets.

Lemma 3.36. *Let $\mathcal{X} \subset \mathbb{R}^d$ be any compact and convex set and let $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$ be any continuous function with (c_α, α)-regular level sets (Condition 3.8). Then, for any radius $r \in (0, \max_{x \in \mathcal{X}} \|x^\star - x\|_2)$, we have that*

$$\mathcal{X} \cap B(x^\star, (r/c_\alpha)^{1+\alpha}) \subseteq \{x \in \mathcal{X} : f(x) \geq \min_{x_r \in \mathcal{S}_r} f(x_r)\} \subseteq B(x^\star, c_\alpha \cdot r^{1/(1+\alpha)})$$

where $\mathcal{S}_r = \{x \in \mathcal{X} : \|x^\star - x\|_2 = r\}$.

Proof. We start to prove the second inclusion. We first show that for all $y \in [\min_{x\in\mathcal{S}_r} f(x), f(x^\star)]$, there exists $x_y \in f^{-1}(y) := \{x \in \mathcal{X} : f(x) = y\}$ such that $\|x^\star - x_y\| \leq r$. Consider any $y \in [\min_{x\in\mathcal{S}_r} f(x), f(x^\star)]$, pick any $x_r \in \arg\min_{x\in\mathcal{S}_r} f(x)$ and introduce the function $F : [0,1] \to \mathbb{R}$ defined by

$$F : \lambda \mapsto f((1-\lambda)x^\star + \lambda x_r),$$

which returns the value of the function f over the segment $[x^\star, x_r]$. As (i) F is continuous, (ii) $F(0) = f(x^\star)$, (iii) $F(1) = \min_{x\in\mathcal{S}_r} f(x)$ and (iv) $y \in [F(1), F(0)]$, it follows from the intermediate value theorem that there exists $\lambda_y \in [0,1]$ such that $F_{x_r}(\lambda_y) = y$. As a consequence, we deduce that there exists $x_y = \lambda_y x^\star + (1-\lambda_y)x_r \in f^{-1}(y)$ satisfying $\|x^\star - x_y\|_2 \leq \|x^\star - x_r\|_2 = r$. Keeping this statement in mind, we may now prove the inclusion. Assume by contradiction that there exists $x'_y \in f^{-1}(y)$ such that $\|x^\star - x'_y\|_2 > c_\alpha r^{1/(1+\alpha)}$. Then, it would follow from the definition of the maximum that $\max_{x\in f^{-1}(y)} \|x^\star - x\|_2 \geq \|x^\star - x'_y\|_2 > c_\alpha r^{1/(1+\alpha)}$. However, as $c_\alpha \cdot \min_{x\in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)} \leq c_\alpha \cdot \|x^\star - x_y\|_2^{1/(1+\alpha)} \leq c_\alpha \cdot r^{1/(1+\alpha)}$ by definition of the minimum, we would get the following contradiction by combining the previous statements with the regularity of the level set of the function:

$$\max_{x\in f^{-1}(y)} \|x^\star - x\|_2 \leq c_\alpha \cdot \min_{x\in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)} < \max_{x\in f^{-1}(y)} \|x^\star - x\|_2 .$$

Since this contradiction holds true for all $y \in [\min_{x\in S_r} f(x_r), f(x^\star)]$, we deduce that $\{x \in \mathcal{X} : f(x) \geq \min_{x\in S_r} f(x_r)\} \subseteq B(x^\star, c_\alpha \cdot r^{1/(1+\alpha)})$, which proves the second inclusion. We use similar arguments to prove the first inclusion. Suppose by contradiction that there exists $x' \in \mathcal{X} \cap B(x^\star, (r/c_\alpha)^{1+\alpha})$ such that $f(x) < f(x_r)$ and introduce the function $F : \lambda \in [0,1] \mapsto f((1-\lambda)x^\star + \lambda x')$. Again, we know from the intermediate value theorem that there exists $x'_r \in f^{-1}(f(x_r))$ such that $\|x^\star - x'_r\|_2 < (r/c_\alpha)^{1+\alpha}$. Hence, we have $c_\alpha \cdot \min_{x\in f^{-1}(f(x_r))} \|x^\star - x\|_2^{1/(1+\alpha)} \leq c_\alpha \|x^\star - x'_r\|_2^{1/(1+\alpha)} < r$. However, as $\max_{x\in f^{-1}(f(x_r))} \|x^\star - x\|_2 \geq \|x^\star - x_r\|_2 = r$, we get a similar contradiction as the one previously obtained, which proves the first inclusion. □

3.7.2 Proofs of Section 3.3

Here, we provide the proofs of Proposition 3.14, Corollary 3.15 and Theorem 3.16.

Proof of Proposition 3.14. The statement is proved by induction. Since $X_1 \sim \mathcal{U}(\mathcal{X})$, the result trivially holds for $n = 1$. Suppose now that the statement holds for a given $n \geq 1$, let $X_1, \ldots, X_{n+1}$ be a sequence of evaluation points generated by RankOpt tuned with $\mathcal{R}$ over f after $n+1$ iterations and let $X'_1, \ldots, X'_{n+1}$ be a sequence of $n+1$ independent points randomly sampled over $\mathcal{X}$. Pick any $y \in \mathbb{R}$ and let $\mathcal{X}_y = \{x \in \mathcal{X} : f(x) \geq y\}$ be the corresponding level set. In the sequel, we consider without loss of generality that $\mu(\mathcal{X}_y) > 0$ (otherwise $\mathbb{P}(\max_{i=1\ldots n+1} f(X_i) \geq y) = 0$ and the result would trivially hold). Denoting by $\mathcal{X}_n = \{x \in \mathcal{X} : \exists r \in \mathcal{R}_n \text{ s.t. } r(x, X_{\hat{\imath}_n}) \geq 0\}$ the sampling area of $X_{n+1}|X_1, \ldots X_n$ and

observing that $\mathcal{X}_y \subseteq \mathcal{X}_n \subseteq \mathcal{X}$ on the event $\{\max_{i=1\dots n} f(X_i) < y\}$, one can directly write

$$\begin{aligned}
\mathbb{P}\left(\max_{i=1\dots n+1} f(X_i) \geq y\right) &= \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i) < y,\ X_{n+1} \in \mathcal{X}_y\right\}\right] \\
&= \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i) < y\right\} \frac{\mu(\mathcal{X}_n \cap \mathcal{X}_y)}{\mu(\mathcal{X}_n)}\right] \\
&\geq \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i) < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})}\right] \\
&\geq \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i') \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i') < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})}\right] \\
&= \mathbb{P}\left(\max_{i=1\dots n+1} f(X_i') \geq y\right)
\end{aligned}$$

where we used on the fourth line the fact that the function $x \mapsto \mathbb{I}\{x \geq y\} + \mathbb{I}\{x < y\}\mu(\mathcal{X}_y)/\mu(\mathcal{X})$ is non-decreasing and that $\max_{i=1\dots n} f(X_i)$ is assumed to be superior or equal to $\max_{i=1\dots n} f(X_i')$ in the usual stochastic ordering sense by induction.

Equipped with Proposition 3.14, we may now easily prove the consistency property.

Proof of Corollary 3.15. Pick any $\varepsilon > 0$ and let $\mathcal{X}_{f^\star-\varepsilon} = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \varepsilon\}$ be the corresponding level set. Denoting by $X_1', \dots X_n'$ a sequence of n independent random variables uniformly distributed over $\mathcal{X}$ and using the fact that $0 < \mu(\mathcal{X}_{f^\star-\varepsilon})/\mu(\mathcal{X}) \leq 1$ by Condition 3.7 and Proposition 3.14, we directly get that $\forall n \in \mathbb{N}^\star$,

$$\begin{aligned}
\mathbb{P}\left(\max_{i=1\dots n} f(X_i) < \max_{x\in\mathcal{X}} f(x) - \varepsilon\right) &\leq \mathbb{P}\left(\max_{i=1\dots n} f(X_i') < \max_{x\in\mathcal{X}} f(x) - \varepsilon\right) \\
&= \mathbb{P}\left(X_1' \notin \mathcal{X}_{f^\star-\varepsilon}\right)^n \\
&= \left(1 - \frac{\mu(\mathcal{X}_{f^\star-\varepsilon})}{\mu(\mathcal{X})}\right)^n \\
&\underset{n\to\infty}{\longrightarrow} 0
\end{aligned}$$

which proves the result. □

We may now turn to the proof of the upper bound.

Proof of Theorem 3.16. Observe first that since $r_f \in \mathcal{R} \subseteq \mathcal{R}_\infty$ is a continuous ranking rule, we know from Proposition 3.2 that there exists a continuous function $h \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$ which shares the same ranking rule with f. Since all the arguments used in the proof only use function comparisons, one can then assume without loss of generality that $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$. Moreover, since the result trivially holds whenever the upper bound of the theorem, denoted here by $r_{\delta,n}$, satisfies $r_{\delta,n} \geq \max_{x\in\mathcal{X}} \|x - x^\star\|_2$, we consider the case where $r_{\delta,n} < \max_{x\in\mathcal{X}} \|x - x^\star\|_2$ which necessarily implies by the level set assumption that $\ln(1/\delta) < n$. We also set some notations: set $\mathcal{S}_{\delta,n} = \{x \in \mathcal{X} : \|x^\star - x\|_2 = (r_{\delta,n}/c_\alpha)^{1+\alpha}\}$

and let $R_{\delta,n} = ((r_{\delta,n}/c_\alpha)^{1+\alpha}/c_\alpha)^{1+\alpha}$. We are now ready to prove the result. Denoting by $X'_1,\dots,X'_n$ a sequence of n independent points uniformly distributed over $\mathcal{X}$ and sequentially using Lemma 3.36, Proposition 3.14, Lemma 3.36 and Lemma A.4 gives that

$$\begin{aligned}
\mathbb{P}(\|X_{\hat{\imath}_n} - x^\star\|_2 \leq r_{\delta,n}) &= \mathbb{P}(X_{\hat{\imath}_n} \in B(x^\star, r_{\delta,n})) \\
&\geq \mathbb{P}\left(\max_{i=1\dots n} f(X_i) \geq \min_{x\in\mathcal{S}_{\delta,n}} f(x)\right) && \text{(Lem. 33)} \\
&\geq \mathbb{P}\left(\max_{i=1\dots n} f(X'_i) \geq \min_{x\in\mathcal{S}_{\delta,n}} f(x)\right) && \text{(Prop. 3.14)} \\
&= \mathbb{P}\left(\bigcup_{i=1}^{n} \{X'_i \in \mathcal{X} \cap B(x^\star, R_{\delta,n})\}\right) \\
&= 1 - \left(1 - \frac{\mu(\mathcal{X} \cap B(x^\star, R_{\delta,n}))}{\mu(\mathcal{X})}\right)^n && (X'_i \sim \mathcal{U}(\mathcal{X})) \\
&\geq 1 - \left(1 - \frac{R_{\delta,n}}{\operatorname{diam}(\mathcal{X})}\right)^n && \text{(Lem. A.4)} \\
&\geq 1 - \left(1 - \frac{\ln(1/\delta)}{n}\right)^n && \text{(def. of } R_{\delta,n}) \\
&\geq 1 - \delta && (1 - x \leq e^{-x})
\end{aligned}$$

which proves the result. □

To prove Theorem 3.19, we start by developing the full proof for Proposition 3.18 and we provide two technical lemmas (Lemma 3.37 and Lemma 3.38).

Proof of Proposition 3.18. Again, the statement is proved by induction. Since X_1 and $X_1^\star$ are uniformly distributed over $\mathcal{X}$, the result directly holds for $n = 1$. Suppose now that the statement holds for a given $n \geq 1$, let $X_1,\dots,X_{n+1}$ be a sequence of evaluation points generated by RankOpt tuned with $\mathcal{R}$ over f after $n+1$ iterations and let $X_1^\star,\dots,X_{n+1}^\star$ be a sequence of $n+1$ points distributed as a Pure Adaptive Search indexed by f over $\mathcal{X}$. Pick any $y \in \mathbb{R}$ and let $\mathcal{X}_y = \{x \in \mathcal{X} : f(x) \geq y\}$ be the corresponding level set. In the sequel, we consider without loss of generality that $\mu(\mathcal{X}_y) > 0$ (otherwise $\mathbb{P}(\max_{i=1\dots n+1} f(X_i) \geq y) = 0$ and the result would trivially hold). Denoting by $\mathcal{X}_n = \{x \in \mathcal{X} : \exists r \in \mathcal{R}_n \text{ s.t. } r(x, X_{\hat{\imath}_n}) \geq 0\}$ the sampling area of $X_{n+1}|X_1,\dots X_n$ and observing that $\mathcal{X}_y \subseteq \mathcal{X}_{f(X_{\hat{\imath}_n})} := \{x \in \mathcal{X} : f(x) \geq \max_{i=1\dots n} f(X_i)\} \subseteq \mathcal{X}_n$ on the event $\{\max_{i=1\dots n} f(X_i) < y\}$, we directly get that

$$\begin{aligned}
\mathbb{P}\left(\max_{i=1\dots n+1} f(X_i) \geq y\right) &= \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i) < y\right\} \frac{\mu(\mathcal{X}_n \cap \mathcal{X}_y)}{\mu(\mathcal{X}_n)}\right] \\
&\leq \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i) < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X}_{f(X_{\hat{\imath}_n})})}\right] \\
&\leq \mathbb{E}\left[\mathbb{I}\left\{\max_{i=1\dots n} f(X_i^\star) \geq y\right\} + \mathbb{I}\left\{\max_{i=1\dots n} f(X_i^\star) < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X}_{f(X_{\hat{\imath}_n})})}\right] \\
&= \mathbb{P}\left(\max_{i=1\dots n+1} f(X_i^\star) \geq y\right)
\end{aligned}$$

where we used the fact that the function $x \in \mathcal{X} \mapsto \mathbb{I}\{x \geq y\} + \mathbb{I}\{x < y\}\, \mu(\mathcal{X}_y)/\mu(\mathcal{X}_x)$ is non-decreasing and that $\max_{i=1\dots n} f(X_i)$ is inferior or equal to $\max_{i=1\dots n} f(X_i^\star)$ in the usual stochastic ordering sense (induction assumption).

The next lemma will be used in the proof of Theorem 3.19 to control the volume of the level set of the highest value observed by a Pure Adaptive Search.

Lemma 3.37. *Let $\mathcal{X} \subset \mathbb{R}^d$ be any compact and convex set with non-empty interior, let $f : \mathcal{X} \to \mathbb{R}$ be any function such that $r_f \in \mathcal{R}_\infty$ and let $X_1^\star \dots X_n^\star$ be a sequence of n random variables distributed as a Pure Adaptive Search indexed by f over $\mathcal{X}$. Then, for any $u \in [0,1]$, we have that*

$$\mathbb{P}\left(\frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq u\right) \leq \mathbb{P}\left(\prod_{i=1}^{n} U_i \leq u\right)$$

where $\mathcal{X}_n^\star := \{x \in \mathcal{X} : f(x) \geq f(X_n^\star)\}$ and $U_1, \dots, U_n$ are n independent random variables uniformly distributed over $[0,1]$.

Proof. First, observe that if $u^\star = \mu(\{x \in \mathcal{X} : f(x) \geq \max_{x \in \mathcal{X}} f(x)\})/\mu(\mathcal{X}) > 0$, then the result trivially holds for any $u < u^\star$ and $n \geq 1$. We thus consider for simplicity that $u^\star = 0$. We also set some notations: $\forall u \in [0,1]$, set $y_u := \min\{y \in \mathrm{Im}(f) : \mu(\{x \in \mathcal{X} : f(x) \geq y\})/\mu(\mathcal{X}) \leq u\}$ and let $\mathcal{X}_{y_u} = \{x \in \mathcal{X} : f(x) \geq y_u\}$ be the corresponding level set. Keeping in mind that $\mu(\mathcal{X}_{y_u}) \leq u \cdot \mu(\mathcal{X})$ for all $u \in [0,1]$, we may now prove the result by induction.

Set $n = 1$, pick any $u \in [0,1]$ and let $U_1 \sim \mathcal{U}([0,1])$. As $X_1^\star \sim \mathcal{U}(\mathcal{X})$ and $\mathbb{P}(U_1 \leq u) = u$, it directly follows that

$$\mathbb{P}\left(\frac{\mu(\mathcal{X}_1^\star)}{\mu(\mathcal{X})} \leq u\right) = \mathbb{P}(X_1^\star \in \mathcal{X}_{y_u}) = \frac{\mu(\mathcal{X}_{y_u})}{\mu(\mathcal{X})} \leq u = \mathbb{P}(U_1 \leq u)$$

Assume now that the statement holds for a given $n \in \mathbb{N}^\star$, fix any $u \in [0,1]$, let $X_1^\star, \dots, X_{n+1}^\star$ be a sequence of $n+1$ random variables distributed as Pure Adaptive Search indexed by f over $\mathcal{X}$ and let $U_1, \dots, U_{n+1}$ be sequence of $n+1$ independent random variables uniformly distributed over $[0,1]$, also independent of $X_i^\star$, $i \geq 1$. Denoting by $\mathcal{X}_n^\star = \{x \in \mathcal{X} : f(x) \geq f(X_n^\star)\}$ the sampling area of $X_{n+1}^\star | \, X_n^\star$ and observing that the

following events are equivalent $\{\mu(\mathcal{X}_n^\star) > \mu(\mathcal{X}_{y_u})\} = \{\mathcal{X}_n^\star \subset \mathcal{X}_{y_u}\}$, we obtain that:

$$\begin{aligned}
\mathbb{P}\left(\frac{\mu(\mathcal{X}_{n+1}^\star)}{\mu(\mathcal{X})} \leq u\right) &= \mathbb{E}\left[\mathbb{I}\{\mu(\mathcal{X}_n^\star) \leq \mu(\mathcal{X}_{y_u})\} + \mathbb{I}\{\mu(\mathcal{X}_n^\star) > \mu(\mathcal{X}_{y_u})\} \cdot \frac{\mu(\mathcal{X}_{y_u} \cap \mathcal{X}_n^\star)}{\mu(\mathcal{X}_n^\star)}\right] \\
&= \mathbb{E}\left[\mathbb{I}\{\mu(\mathcal{X}_n^\star) \leq \mu(\mathcal{X}_{y_u})\} + \mathbb{I}\{\mu(\mathcal{X}_n^\star) > \mu(\mathcal{X}_{y_u})\} \cdot \frac{\mu(\mathcal{X}_{y_u})}{\mu(\mathcal{X}_n^\star)}\right] \\
&= \mathbb{E}\left[\min\left(1, \frac{\mu(\mathcal{X}_{y_u})}{\mu(\mathcal{X}_n^\star)}\right)\right] \\
&= \mathbb{E}\left[\mathbb{P}\left(U_{n+1} \leq \min\left(1, \frac{\mu(\mathcal{X}_{y_u})}{\mu(\mathcal{X}_n^\star)}\right) \mid \mu(\mathcal{X}_n^\star)\right)\right] \\
&= \mathbb{P}\left(U_{n+1} \cdot \frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq \frac{\mu(\mathcal{X}_{y_u})}{\mu(\mathcal{X})}\right) \\
&\leq \mathbb{P}\left(U_{n+1} \cdot \frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq u\right) \\
&\leq \mathbb{P}\left(\prod_{i=1}^{n+1} U_i \leq u\right).
\end{aligned}$$

and the proof is complete. □

The concentration inequality provided in the next lemma will be important in order to control the volume of the level set of the highest value observed by a Pure Adaptive Search.

Lemma 3.38. *Let $U_1, \ldots, U_n$ be a sequence of n independent random variables uniformly distributed over $[0,1]$. Then, for any $\delta \in (0,1)$, we have that $\mathbb{P}\left(\prod_{i=1}^n U_i < \delta \cdot e^{-n-\sqrt{2n\ln(1/\delta)}}\right) < \delta$.*

Proof. Taking the logarithm on both sides gives that $\prod_{i=1}^n U_i < \delta \cdot e^{-n-\sqrt{2n\ln(1/\delta)}}$ if and only if $\sum_{i=1}^n -\ln(U_i) > n + \sqrt{2n\ln(1/\delta)} + \ln(1/\delta)$. As $U_i \sim \mathcal{U}([0,1])$ for $i \leq n$, we have that $-\ln(U_i) \sim \text{Exp}(1)$ which combined with independence gives that $\sum_{i=1}^n -\ln(U_i) \sim \text{Gamma}(n,1)$. Therefore, the desired result follows from the application of a standard concentration inequality for sub-gamma random variables (see Chapter 2.4 in Boucheron et al. (2013)). □

Equipped with Proposition 3.18, Lemma 3.37 and Lemma 3.38, we may now prove the lower bound.

Proof of Theorem 3.19. Similarly to the proof of Theorem 3.16, we consider without loss of generality that $f \in \mathcal{C}^0(\mathcal{X}, \mathbb{R})$. Fix any $n \in \mathbb{N}^\star$ and any $\delta \in (0,1)$, let $r_{\delta,n}$ be the lower bound of the theorem, set $\mathcal{S}_{\delta,n} = \{x \in \mathcal{X} : \|x^\star - x\|_2 = c_\alpha r_{\delta,n}^{1/(1+\alpha)}\}$ and $R_{\delta,n} = \text{rad}(\mathcal{X})\,\delta^{1/d}\exp(-(n + \sqrt{2n\ln(1/\delta)})/d)$. Denoting by $X_1^\star, \ldots, X_n^\star$ a sequence of n random variables distributed as Pure Adaptive Search indexed by f over $\mathcal{X}$ and by $\mathcal{X}_n^\star = \{x \in \mathcal{X} : f(x) \geq f(X_n^\star)\}$ the level set of $f(X_n^\star)$, successively using the first inclu-

sion of Lemma 3.36, Proposition 3.18, Lemma 3.36, 3.37 and 3.38, we obtain:

$$
\begin{aligned}
\mathbb{P}(\|X_{\hat{\imath}_n} - x^\star\|_2 \leq r_{\delta,n}) &= \mathbb{P}(X_{\hat{\imath}_n} \in B(x^\star, r_{\delta,n}) \cap \mathcal{X}) \\
&\leq \mathbb{P}\left(f(X_{\hat{\imath}_n}) \geq \min_{x \in \mathcal{S}_{\delta,n}} f(x)\right) && \text{(Lem. 3.36)} \\
&\leq \mathbb{P}\left(f(X_n^\star) \geq \min_{x \in \mathcal{S}_{\delta,n}} f(x)\right) && \text{(Prop. 3.18)} \\
&= \mathbb{P}\left(\mu(\mathcal{X}_n^\star) \leq \mu\left(\left\{x \in \mathcal{X} : f(x) \geq \min_{x \in \mathcal{S}_{\delta,n}} f(x)\right\}\right)\right) \\
&\leq \mathbb{P}\left(\frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq \frac{\mu(B(x^\star, R_{\delta,n}))}{\mu(\mathcal{X})}\right) && \text{(Lem. 3.36)} \\
&\leq \mathbb{P}\left(\frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq \frac{\mu(B(x^\star, R_{\delta,n}))}{\mu(B(x, \mathrm{rad}(\mathcal{X})))}\right) \\
&= \mathbb{P}\left(\frac{\mu(\mathcal{X}_n^\star)}{\mu(\mathcal{X})} \leq \left(\frac{R_{\delta,n}}{\mathrm{rad}(\mathcal{X})}\right)^d\right) \\
&\leq \mathbb{P}\left(\prod_{i=1}^n U_i \leq \delta \cdot e^{-n-\sqrt{2n\ln(1/\delta)}}\right) && \text{(Lem. 3.37 and def. of } R_{\delta,n}) \\
&< \delta && \text{(Lem. 3.38)}
\end{aligned}
$$

where $U_1, \ldots, U_n$ denotes a sequence of n i.i.d. copies of $U \sim \mathcal{U}([0,1])$ and we used on the sixth and seventh line the fact that since $\mathrm{rad}(\mathcal{X}) > 0$ is assumed to be finite, then there exists $x \in \mathcal{X}$ such that $B(x, \mathrm{rad}(\mathcal{X})) \subseteq \mathcal{X}$ which implies that $\mu(\mathcal{X}) \geq \mu(B(x, \mathrm{rad}(\mathcal{X}))) = \pi^{d/2}\mathrm{rad}(\mathcal{X})^d / \Gamma(d/2+1)$. □

3.7.3 Proofs of Section 3.4

We develop here the proofs of Proposition 3.23, Proposition 3.25 and Theorem 3.26. The proof of the consistency property of the algorithm is straightforward.

Proof of Proposition 3.23. Pick any $\varepsilon > 0$ and let $\mathcal{X}_{f^\star - \varepsilon} = \{x \in \mathcal{X} : f(x) \geq \max_{x \in \mathcal{X}} f(x) - \varepsilon\}$ be the corresponding level set. Since $p \in (0,1)$ and $0 < \mu(\mathcal{X}_{f^\star - \epsilon})/\mu(\mathcal{X}) \leq 1$ by Condition 3.7, it can easily be shown by induction that $\forall n \in \mathbb{N}^\star$:

$$
\mathbb{P}\left(f(X_{\hat{\imath}_n}) < \max_{x \in \mathcal{X}} f(x) - \varepsilon\right) \leq \left(1 - p \cdot \frac{\mu(\mathcal{X}_{f^\star - \varepsilon})}{\mu(\mathcal{X})}\right)^n \xrightarrow[n \to \infty]{} 0
$$

and proves the result. □

We now prove the stopping time upper bound.

Proof of Proposition 3.25. Let $\{(X_i, B_i)\}_{i \in \mathbb{N}^\star}$ be the sequence of random variables defined in the AdaRankOpt algorithm. Fix any $\delta \in (0,1)$ and set $n'_\delta = \lfloor p \cdot n_\delta - \sqrt{n_\delta \log(2/\delta)/2} \rfloor$ where $n_\delta = \lfloor 10 \cdot (K + \log(2/\delta))/(p \cdot \inf_{r \in \mathcal{R}_{N^\star - 1}} L(r)^2) \rfloor$ denotes the integer part of the upper bound. First, observe that since $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \cdots \subset \mathcal{R}_\infty$ forms a

nested sequence,

$$\min_{r\in\mathcal{R}_1} L_{n_\delta}(r) \leq \min_{r\in\mathcal{R}_2} L_{n_\delta}(r) \leq \cdots \leq \min_{r\in\mathcal{R}_{k^\star-1}} L_{n_\delta}(r)$$

where L_{n_δ} denotes the empirical ranking loss taken over the first n_δ samples $\{X_i\}_{i=1}^{n_\delta}$. One might then start with the following decomposition:

$$\begin{aligned}\mathbb{P}(\tau_{k^\star} \leq n_\delta) &= \mathbb{P}\left(\min_{r\in\mathcal{R}_{k^\star-1}} L_{n_\delta}(r) > 0\right) \\ &\geq \mathbb{P}\left(\min_{r\in\mathcal{R}_{k^\star-1}} L_{n_\delta}(r) > 0 \mid \sum_{i=1}^{n_\delta} B_i \geq n'_\delta\right) \times \mathbb{P}\left(\sum_{i=1}^{n_\delta} B_i \geq n'_\delta\right). \end{aligned} \tag{3.2}$$

We now focus on the first term of the right hand side of (3.2) and we start to lower bound the empirical risk by only keeping the first n'_δ i.i.d. exploratory samples:

$$L_{n_\delta}(r) \geq \frac{2}{n_\delta(n_\delta-1)} \sum_{1\leq i<j\leq n_\delta} \mathbb{I}\{r(X_i,X_j) \neq r_f(X_i,X_j)\} \times \mathbb{I}\{(i,j)\in I^2\} \tag{3.3}$$

where $I = \{i \leq n_\delta : B_i = 1 \text{ and } \sum_{j=1}^{i} B_i \leq n'_\delta\}$. By definition **??**, we know that conditioned upon $|I|$, the sequence $\{X_i\}_{i\in I}||I|$ is a sequence of $|I|$ independent random variables uniformly distributed over $\mathcal{X}$. Therefore, conditioned upon the event $\{\sum_{i=1}^{n_\delta} B_i \geq n'_\delta\} = \{|I| = n'_\delta\}$, the right hand term of (3.3) has the same distribution as

$$\frac{2}{n_\delta(n_\delta-1)} \sum_{1\leq i<j\leq n_{\delta'}} \mathbb{I}\Big\{r(X'_i,X'_j) \neq r_f(X'_i,X'_j)\Big\} \propto L_{n'_\delta}(r)$$

where the sequence $\{X'_i\}_{i=1}^{n'_\delta} \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$ is independent of $\{B_i\}_{i=1}^{n_\delta}$. Hence, we have that

$$\mathbb{P}\left(\min_{r\in\mathcal{R}_{k^\star-1}} L_{n_\delta}(r) > 0 \mid \sum_{i=1}^{n_\delta} B_i \geq n'_\delta\right) \geq \mathbb{P}\left(\min_{r\in\mathcal{R}_{k^\star-1}} L_{n'_\delta}(r) > 0\right)$$

where $L_{n'_\delta}$ denotes the empirical ranking loss is taken over $\{X'_i\}_{i=1}^{n'_\delta}$. Now, by slightly adapting the generalization bound on bipartite ranking rules of **?** (*i.e.,* see the proof of Corollary 3 in Section 3 therein) we obtain that with probability at least $1-\delta/2$,

$$\sup_{r\in\mathcal{R}_{k^\star-1}} \Big|L_{n'_\delta}(r) - L(r)\Big| \leq 2\mathbb{E}\left[R_{n'_\delta}(\mathcal{R}_{k^\star-1})\right] + 2\sqrt{\frac{\log(2/\delta)}{n'_\delta - 1}},$$

which combined with the fact that $\mathbb{E}\left[R_{n'_\delta}(\mathcal{R}_{k^\star-1})\right] \leq \sqrt{K/n}$ gives that

$$\min_{r\in\mathcal{R}_{k^\star-1}} L_{n'_\delta}(r) \geq \inf_{r\in\mathcal{R}_{k^\star-1}} L(r) - 2\sqrt{\frac{K}{n'_\delta}} - 2\sqrt{\frac{\log(2/\delta)}{n'_\delta - 1}}.$$

Finally, as n'_δ and n_δ were defined (with express purpose) so that (i) the right hand term of the previous inequality is strictly positive and so that (ii) Hoeffding's inequality ensures that

$$\mathbb{P}\left(\sum_{i=1}^{n_\delta} B_i \geq n'_\delta\right) \geq 1-\delta/2,$$

we deduce from (3.2) that $\mathbb{P}(\tau_{k^\star} \leq n_\delta) \geq (1-\delta/2)^2 \geq 1-\delta$ and the proof is complete. □

Theorem 3.26 is obtained by combining the upper bounds of Proposition 3.25 and Theorem 3.16.

Proof of Theorem 3.26. Fix $\delta \in (0,1)$, let $n_{\delta/2} = \lfloor 10(K+\ln(4/\delta))/(p \cdot \inf_{r\in\mathcal{R}_{k^\star-1}} L(r)^2) \rfloor$ be the integer part of the upper bound of Proposition 3.25 (set with probability $1-\delta/2$) and let $r_{\delta/2,n}$ be the upper bound of the Theorem 3.16 (also set with probability $1-\delta/2$). We use the following decomposition:

$$\mathbb{P}(\|X_{\hat{i}_n} - x^\star\|_2 \leq r_{\delta/2,n}) \geq \mathbb{P}(\|X_{\hat{i}_n} - x^\star\|_2 \leq r_{\delta/2,n} \,|\, \tau_{k^\star} < n_{\delta/2}) \times \mathbb{P}(\tau_{k^\star} < n_{\delta/2}). \tag{3.4}$$

First, as on the event $\{\tau_{k^\star} < n_{\delta/2}\} = \bigcap_{t\geq n_{\delta/2}} \{\widehat{k}_t = k^\star\}$ the smallest ranking structure $\mathcal{R}_{k^\star}$ containing the true ranking rule r_f is identified for all $t \geq n_{\delta/2}$, one can easily check that $\mathbb{P}(\|X_{\hat{i}_n} - x^\star\|_2 \leq r_{\delta/2,n} \mid \tau_{k^\star} \leq r_{\delta,n}) \geq 1-\delta/2$ by reproducing the same steps as in Theorem 3.16's proof with the last $n - n_{\delta/2}$ samples. Second, as Proposition 3.25 also guarantees that $\mathbb{P}(\tau_{k^\star} < n_{\delta/2}) \geq 1-\delta/2$, we then obtain from (3.4) that $\mathbb{P}(\|X_{\hat{i}_n} - x^\star\|_2 \leq r_{\delta/2,n}) \geq (1-\delta/2)^2 \geq 1-\delta$. Hence, for all $n > n_{\delta/2}$, we have with probability at least $1-\delta$,

$$\begin{aligned}
\|X_{\hat{i}_n} - x^\star\|_2 &\leq C_1 \cdot \left(\frac{\ln(2/\delta)}{n - n_{\delta/2}}\right)^{\frac{1}{d(1+\alpha)^2}} \\
&= C_1 \cdot \left(1 + \frac{n_{\delta/2}}{n - n_{\delta/2}}\right)^{\frac{1}{d(1+\alpha)^2}} \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}} \\
&\leq C_1 \cdot \left(\frac{11(K+\ln(4/\delta))}{p \inf_{r\in\mathcal{R}_{k^\star-1}} L(r)^2}\right) \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{1}{d(1+\alpha)^2}}
\end{aligned}$$

which proves the result since the right hand term of the previous inequality is superior or equal to $\mathrm{diam}(\mathcal{X})$ whenever $n \leq n_{\delta/2}$. □

3.7.4 Proofs of Section 3.5

We state here the proofs of Proposition 3.29, Lemma 3.30, Corollary 3.31, Proposition 3.34 and Proposition 3.35. We start with the proofs of Proposition 3.29, Lemma 3.30 and Proposition 3.35.

Proof of Proposition 3.29 ($\Rightarrow$) Assume that there exists $r \in \mathcal{R}_{\mathcal{P}(k)}$ such that $L_{t+1}(r) = 0$. By definition of $\mathcal{R}_{\mathcal{P}(k)}$, we know that there exists a polynomial function f_r of degree k such that $\forall (x,x') \in \mathcal{X}^2$, $r(x,x') = \mathrm{sgn}(f(x) - f(x'))$. Moreover, as $f_r \in \mathcal{P}_k(\mathcal{X},\mathbb{R})$, there exists some $(\omega_r, c_r) \in \mathbb{R}^{\dim(\Phi_k)} \times \mathbb{R}$ such that $\forall x \in \mathbb{R}$, $f_r(x) = \langle \omega_r, \Phi_k(x)\rangle + c_r$. Hence, putting the previous statements altogether with the fact that $L_{t+1}(r) = 0$ gives that $\forall i \leq t$,

$$1 = r(X_{(i+1)}, X_{(i)}) = \mathrm{sgn}(f_r(X_{(i+1)}) - f_r(X_{(i)})) = \mathrm{sgn}(\langle \omega_r, \Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)})\rangle)$$

and proves that there exists $\omega = \omega_r \in \mathbb{R}^{\dim(\Phi_k)}$ such that for all $i \leq t$, $\langle \omega, \Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)})\rangle > 0$

($\Leftarrow$) Assume now that there exists $\omega \in \mathbb{R}^{\dim(\Phi_k)}$ such that $\forall i \in \{1 \dots t\}$ $\langle \omega, \Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}) \rangle > 0$. Introduce the polynomial function of degree k defined by $f_\omega : x \mapsto \langle \omega, \Phi_k(x) \rangle + c$ where $c \geq 0$ is any arbitrary constant. Now, if r_{f_ω} denotes the polynomial ranking rule induced by f_ω, we obtain from the first assumption that $\forall i \leq t$,

$$r_{f_\omega}(X_{(i+1)}, X_{(i)}) = \text{sgn}(f_\omega(X_{(i+1)}) - f_\omega(X_{(i)})) = \text{sgn}(\langle \omega, \Phi_k(X_{i+1}) - \Phi_k(X_{(i)}) \rangle) = 1.$$

We thus deduce that $L_{t+1}(r) = 0$, which proves that there exists $r = r_{f_w} \in \mathcal{R}_{\mathcal{P}_k}$ such that $L_{t+1}(r) = 0$. □

Proof of Lemma 3.30. First, observe that $Y_i \cdot \langle \omega, X_i \rangle > 0$ if and only if $\langle \omega, Y_i \cdot X_i \rangle > 0$. We thus consider without loss of generality that $Y_i = 1$ for all $i \leq t$, by replacing X_i with $Y_i \cdot X_i$.

($\Rightarrow$) Assume that there exists $\omega \in \mathbb{R}^d$ such that $\forall i \in \{1 \dots t\}$, $\langle \omega, X_i \rangle > 0$. If $\vec{0} \in \text{CH}\{X_i\}_{i=1}^t$, this would mean that there exists $(\lambda_1, \dots, \lambda_t) \in \mathbb{R}^t$ such that (i) $\vec{0} = \sum_{i=1}^t \lambda_i \cdot X_i$, (ii) $\sum_{i=1}^n \lambda_i = 1$ and (iii) $\lambda_i \geq 0$, $i = 1 \dots t$ and it would give us to the following contradiction:

$$0 = \langle \omega, \vec{0} \rangle = \sum_{i=1}^{t} \lambda_i \cdot \langle \omega, X_i \rangle > 0$$

and we deduce that, necessarily, $\vec{0} \notin \text{CH}\{X_i\}_{i=1}^t$.

($\Leftarrow$) Assume now that $\vec{0} \notin \text{CH}\{X_i\}_{i=1}^t$. Since t and d are finite, $\text{CH}\{X_i\}_{i=1}^t$ is a closed, compact and convex set and $\min_{x \in \text{CH}\{X_i\}_{i=1}^t} \|x\|_2 = d_{\min}$ exists. Moreover, the condition $\vec{0} \notin \text{CH}\{X_i\}_{i=1}^t$ implies that $d_{\min} > 0$. Now, let $x_d \in \text{CH}\{X_i\}_{i=1}^t$ be the (unique) point of the convex hull satisfying $\|x_d\|_2 = d_{\min}$. We prove by contradiction that $\forall x \in \text{CH}\{X_i\}_{i=1}^t$, $\langle x, x_d \rangle \geq d_{\min}^2$. Suppose that there exists $x \in \text{CH}\{X_i\}_{i=1}^t$ such that $\langle x, x_d \rangle < {d_{\min}}^2$. First, we know from the convexity of the convex hull that the whole line $L = (x, x_d)$ also belongs to $\text{CH}\{X_i\}_{i=1}^t$. However, since $\|x_d\|_2 = d_{\min}$ and $\langle x, x_d \rangle < \|x_d\|^2$, the line L is not tangent to the ball $B(\vec{0}, d_{\min})$ and intersects it. Therefore, there necessarily exists some $x' \in L \cap B(\vec{0}, d_{\min})$ such that $\|x'\|_2 < d_{\min}$. However, as $x' \in L \subseteq \text{CH}\{X_i\}_{i=1}^t$ also belongs to the convex hull, we obtain the following contradiction:

$$\min_{x \in \text{CH}\{X_i\}_{i=1}^t} \|x\|_2 \leq \|x'\|_2 < d_{\min} = \min_{x \in \text{CH}\{X_i\}_{i=1}^t} \|x\|_2$$

and we deduce that $\forall x \in \text{CH}\{X_i\}_{i=1}^t$, $\langle x_d, x \rangle \geq d_{\min} > 0$. Finally, since $\{X_i\}_{i=1}^t \in \text{CH}\{X_i\}_{i=1}^t$, we thus deduce that there exists some $\omega = x_d \in \mathbb{R}^d$ such that $\forall i \in \{1 \dots t\}$, $\langle \omega, X_i \rangle > 0$. □

Corollary 3.31 is obtained by combining Proposition 3.29 with Lemma 3.30.

Proof of Corollary 3.31 From Proposition 3.29, we have the following equivalence:

$$\min_{r \in \mathcal{R}_{\mathcal{P}_k}} L_{t+1}(r) = 0 \;\Leftrightarrow\; \exists \omega \in \mathbb{R}^{\dim(\phi_k)} \text{ s.t. } \langle \omega, \Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}) \rangle > 0,\ \forall i \in \{1 \dots t\}$$

which combined with Lemma 3.30 gives

$$\min_{r\in\mathcal{R}_{\mathcal{P}_k}} L_{t+1}(r) = 0 \;\Leftrightarrow\; \vec{0} \notin \mathrm{CH}\{(\Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}))\}_{i=1}^{t}.$$

In addition, we know from the vertex representation of convex hulls that $\vec{0} \notin \mathrm{CH}\{(\Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)}))\}_{i=1}^{t}$ if and only if there does not exist any $\lambda = (\lambda_1,\ldots,\lambda_t) \in \mathbb{R}^t$ such that (i) $\sum_{i=1}^{t}\lambda_i\,(\Phi_k(X_{(i+1)}) - \Phi_k(X_{(i)})) = \vec{0}$, (ii) $\sum_{i=1}^{t}\lambda_i = 1$ and (iii) $\lambda_i \geq 0$, $i = 1,\ldots,t$ and therefore putting those constraints (i), (ii) and (iii) into matrix form leads us to the desired equivalence:

$$\min_{r\in\mathcal{R}_{\mathcal{P}_k}} L_{t+1}(r) = 0 \;\Leftrightarrow\; \left\{\lambda \in \mathbb{R}^t :\, \mathrm{M}_t^{\Phi_k}\lambda^{\mathsf{T}} = \vec{0},\, \langle\vec{1},\lambda\rangle = 1,\, \lambda \succeq \vec{0}\right\} = \emptyset$$

where $\mathrm{M}_k^{\Phi_k}$ is the matrix defined in the corollary. □

Now, we provide the proofs for Proposition 3.34 and Proposition 3.35.

Proof of Proposition 3.34 ($\Rightarrow$) Assume that there exists $r \in \mathcal{R}_{\mathcal{C}_k}$ such that $L_{t+1}(r) = 0$ and let $\{h_i\}_{i=1}^{t+1}$ be the sequence of classifiers defined $\forall i \leq t+1$ by $h_i(x) = \mathbb{I}\left\{r(x, X_{(i)}) \geq 0\right\}$. First, we know from the definition of $\mathcal{R}_{\mathcal{C}_k}$ that the classifiers h_i are all of the form $h_i(x) = \sum_{m=1}^{k}\mathbb{I}\{l_{i,m} \leq x \leq u_{i,m}\}$. Secondly, as $L_{n+1}(r) = 0$, it necessarily follows that $\forall (i,j) \in \{1,\ldots,t+1\}^2$, $h_i(X_{(j)}) = \mathbb{I}\{j \geq i\}$. Finally, since $r(\cdot,\cdot)$ is transitive and $r(X_{(i+1)}, X_{(i)}) = 1$ for all $i \leq t$, we deduce that $h_1 \geq h_2 \geq \cdots \geq h_{t+1}$.

($\Leftarrow$) Assume now that there exists a sequence of classifiers $\{h_i\}_{i=1}^{t+1}$ of the form $h_i(x) = \sum_{m=1}^{k}\mathbb{I}\{l_{i,m} \leq x \leq u_{i,m}\}$ satisfying: (i) $h_1 \geq h_2 \geq \cdots \geq h_{t+1}$ and (ii) $\forall (i,j) \in \{1\ldots t+1\}^2$, $h_i(X_{(j)}) = \mathbb{I}\{j \geq i\}$. Define the step function $f_{\text{step}}(x) = \sum_{i=1}^{t+1} h_i(x)$ and observe that $L_{t+1}(r_{f_{\text{step}}}) = 0$ since $\forall (j,k) \in \{1\ldots t+1\}^2$,

$$r_{f_{\text{step}}}(X_{(j)}, X_{(k)}) = \operatorname{sgn}\left(\sum_{i=1}^{t+1}\mathbb{I}\{j \geq i\} - \mathbb{I}\{k \geq i\}\right) = \operatorname{sgn}(j-k) = r_f(X_{(j)}, X_{(k)}),$$

To prove the result, we will simply construct a continuous approximation of the function f_{step} which (i) induces a ranking rule which perfectly ranks the sample and (ii) admits level sets which are unions of at most k convex set. Let $\hat{f}_\epsilon : \mathcal{X} \to \mathbb{R}$ be the continuous function defined by $\hat{f}_\epsilon(x) = \sum_{i=1}^{t+1}\sum_{m=1}^{k}\hat{\mathbb{1}}_{\epsilon,l_{i,m},u_{i,m}}(x)$ where $\forall l \leq u$,

$$\hat{\mathbb{1}}_{\epsilon,l,u}(x) = \begin{cases} 1 & \text{if } \; x \in [l,u] \\ 1 - \frac{l-x}{\epsilon} & \text{if } \; x \in [l-\epsilon, l[\\ 1 - \frac{x-u}{\epsilon} & \text{if } \; x \in]u, u+\epsilon] \\ 0 & \text{otherwise.} \end{cases}$$

Observe now that $\forall \epsilon < \min\{|x_1 - x_2| \,:\, x_1 \neq x_2 \in \{X_{(i)}\}_{i=1}^{t+1} \cup \{l_{i,m}\}_{i=1\ldots t+1}^{m=1\ldots k} \cup \{u_{i,m}\}_{i=1\ldots t+1}^{m=1\ldots k}\}$ and $\forall i \leq t$, we have that $\hat{f}_\epsilon(X_{(i)}) = f_{\text{step}}(X_{(i)})$. Hence $L_{t+1}(r_{\hat{f}_\epsilon}) = L_{t+1}(r_{f_{\text{step}}}) = 0$ which proves (i). Moreover, as for any $\epsilon < \min\{|x_1 - x_2| : x_1 \neq x_2 \in$

$\{l_{i,m}\}_{i=1\dots t+1}^{m=1\dots k} \cup \{u_{i,m}\}_{i=1\dots t+1}^{m=1\dots k}\}/2$, the level sets of $\hat{f}_\epsilon$ are by construction a union of at most k segments (convex sets) and (ii) holds true. We then deduce from (i) and (ii) that for ϵ small enough there exists $r = r_{\hat{f}_\epsilon} \in \mathcal{R}_{\mathcal{C}_k}$ such that $L_{t+1}(r) = 0$. □

Proof of Proposition 3.35 ($\Rightarrow$) Assume that there exists $r \in \mathcal{R}_{\mathcal{C}_1}$ such that $L_{t+1}(r) = 0$. Observe first that since $\forall j \neq k \leq t+1, r(X_{(j)}, X_{(k)}) = 2\mathbb{I}\{j > k\} - 1$, we have that $\forall k \leq t$,

i) $\{X_{(i)}\}_{i=k+1}^{t+1} \in \{x \in \mathcal{X} : r(x, X_{(k+1)}) \geq 0\}$;

ii) $X_{(k)} \notin \{x \in \mathcal{X} : r(x, X_{(k+1)}) \geq 0\}$.

Now, pick any $k \in \{1, \dots, t\}$ and recall that, by definition of $\mathcal{R}_{\mathcal{C}_1}$, the level set $\{x \in \mathcal{X} : r(x, X_{(k+1)}) \geq 0\}$ is also a convex set. However, since $\mathrm{CH}\{X_{(i)}\}_{i=k+1}^{t+1}$ is the smallest convex set which contains $\{X_{(i)}\}_{i=k+1}^{t+1}$, we deduce from (i) that $\mathrm{CH}\{X_{(i)}\}_{i=k+1}^{t+1} \subseteq \{x \in \mathcal{X} : r(x, X_{(k+1)}) \geq 0\}$. Hence, combining the previous statement with (ii) gives that $\forall k \leq t$,

$$X_{(k)} \notin \mathrm{CH}\{X_{(i)}\}_{i=k+1}^{t+1}.$$

Finally, using the vertex representation of convex hulls we know that $X_{(t+1-i)} \notin \mathrm{CH}\{X_{(i)}\}_{i=t+2-i}^{t+1}$ if and only if there does not any $\lambda = (\lambda_1, \dots, \lambda_i) \in \mathbb{R}^i$ such that (i) $\sum_{j=1}^{j} \lambda_j \cdot X_{(t+2-j)} = X_{(k)}$, (ii) $\sum_{j=1}^{i} \lambda_j = 1$ and (iii) $\lambda_j \geq 0$, $j = 1 \dots i$ and putting these constraints (i), (ii) and (iii) into matrix form gives the result.

($\Leftarrow$) Assume now that the cascade of polyhedrons is empty. First, by reproducing the same steps as in the first part of the proof, it can easily be checked that $\forall k \leq t$, $X_{(k)} \notin \mathrm{CH}\{X_{(i)}\}_{i=k+1}^{t+1}$, which implies that

$$\mathrm{CH}\{X_{(t+1)}\} \subset \mathrm{CH}\{X_{(i)}\}_{i=t}^{t+1} \subset \cdots \subset \mathrm{CH}\{X_{(i)}\}_{i=1}^{t+1}. \tag{3.5}$$

Now, define the step function $f_{\text{step}} : x \in \mathcal{X} \mapsto \sum_{i=1}^{t+1} \mathbb{I}\{x \in \mathrm{CH}\{X_{(j)}\}_{j=i}^{t+1}\}$ and observe that $L_{t+1}(r_{f_{\text{step}}}) = 0$ by (3.5). To prove the result, we will simply construct a continuous approximation of the function f_{step} which (i) induces a ranking rule that perfectly ranks the sample and (ii) has convex level sets. Let $\hat{f}_\epsilon$ be the continuous function defined by $\hat{f}_\epsilon(x) = \sum_{i=1}^{t+1} \phi_{i,\epsilon}(x)$ where $\forall i \leq t+1$,

$$\phi_{i,\epsilon}(x) = \begin{cases} 1 - \frac{d(x, B(\mathrm{CH}\{X_{(j)}\}_{j=i}^{t+1}, 2(t+1-i)\epsilon))}{\epsilon} & \text{if } d(x, B(\mathrm{CH}\{X_{(j)}\}_{j=i}^{t+1}, 2(t+1-i)\epsilon) \leq \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

Observe now that for any $\epsilon < \min_{i=1\dots t} d(X_{(i)}, \mathrm{CH}\{X_{(j)}\}_{j=i+1}^{t+1})/(2t+2)$, we have that $\forall i \leq t+1$, $\hat{f}_\epsilon(X_{(i)}) = f_{\text{step}}(X_{(i)})$. Hence $L_{t+1}(r_{\hat{f}_\epsilon}) = L_{t+1}(r_f) = 0$, which proves (i). Moreover, we know from Lemma A.5 that for any $\epsilon < \min_{i=1\dots t} d(X_{(i)}, \mathrm{CH}\{X_{(j)}\}_{j=i+1}^{t+1})/(2t+2)$ and any $x \in \mathcal{X}$, the level set $\{x' \in \mathcal{X} : \hat{f}_\epsilon(x') \geq \hat{f}_\epsilon(x)\}$ is a convex set. Hence (ii) holds true and we then deduce from (i) and (ii) that for ϵ small enough there exists $r = r_{\hat{f}_\epsilon} \in \mathcal{R}_{\mathcal{C}_1}$ such that $L_{t+1}(r_{\hat{f}_\epsilon}) = 0$ and the proof is complete. □

4

Numerical experiments

Abstract. This chapter is dedicated to an empirical comparison of nine global optimization implementations. The main algorithms of the thesis are compared with seven state-of-the-art methods developed from various approaches of global optimization. The test bed inclues two series of optimization problems that naturally arise in the tuning of learning algorithms and two series of non-convex problems commonly met in optimization benchmarks. A total of 200,000 optimization instances were solved. As a result, we find that both the algorithms developed in the thesis display competitive results over most problems of the benchmark for medium accuracy targets, but could certainly be improved to greater precision. Consequently, several extensions are proposed at the end of the chapter to improve their performance. Some parts of this work have been published in Malherbe et al. (2016) and Malherbe and Vayatis (2017).

Contents

4.1 Introduction

This chapter addresses the practical problem of sequentially optimizing the output of an unknown and potentially non-convex function over a continuous and bounded set. The setup we consider assumes that the function evaluations are noiseless and that the derivative information as well as a bounds on the Lipschitz constants are unavailable. This problem is of particular interest when evaluating the function requires numerical simulations with significant computational cost or when the objective function does not satisfy the standard properties used in optimization, such as linearity, convexity or differentiability. There is a large number of algorithms based on various heuristics which have been introduced in order to address this problem, such as genetic algorithms (Rechenberg (1971)), Bayesian methods (Kushner (1964)) or partitioning techniques (Jones et al. (1993)). Concurrent with the development of methods, the demand for such tools is growing (Bergstra and Bengio (2012), Bousquet et al. (2017)) and many software implementations for this class of problems have been proposed, such as the NLopt library (Johnson (2014)), the BayesOpt Library (Martinez-Cantin (2014)) or the CMA-ES implementation (Hansen (2006))). The primary purpose of this work is to address the following questions: (i) which solver is more likely to identify the global optimum of a nonconvex optimization problem, (ii) which solver can find a near-optimal solution with few evaluations and (iii) how do the algorithms developed in the thesis compare with state-of-the-art implementations. To answer these questions, we provide an empirical comparison of nine global optimization implementations. The main algorithms of the thesis are compared with seven different methods, developed from various approaches of global optimization. The test bed includes two series of toy optimization problems that naturally arise in the tuning of machine learning algorithms and two series of synthetic problems that are commonly met in optimization benchmarks. The algorithms were tested under the same conditions and compared according to several criteria, including their ability to find near-global solutions for various accuracies. As a result, we find that both the algorithms developed in the thesis display competitive results over most problems of the benchmark for medium accuracy targets, but could certainly be improved for greater precision. Consequently, we propose several extensions to improve their performance at the end of the chapter. The remainder of the chapter is structured as follows. In Section 4.2, we provide a full description of the experimental setup. In Section 4.3, we provide a listing of the results of the experiments. Finally, the results are discussed in Section 4.4.

4.2 Experimental setup

In this section, we describe (i) the algorithms used for comparison, (ii) the test problems of the benchmark and and (iii) the metrics used to assess performance.

4.2.1 Algorithms

Seven different types of algorithms developed from various approaches of global optimization were considered in addition to AdaLIPO (Chapter 2) and AdaRank (Chapter 3):

- **BayesOpt** (Martinez-Cantin (2014)) is a Bayesian optimization algorithm. It uses a distribution over functions to build a surrogate model of the unknown function. The parameters controlling the distribution are estimated along with the optimization.

- **CMA-ES** (Covariance Matrix Adaptation Evolution Strategy, Hansen (2006)) is an evolutionary algorithm. It samples the next evaluation points according to a multivariate normal distribution with mean vector and covariance matrix computed from the previous evaluations.

- **CRS** (Controlled Random Search, Kaelo and Ali (2006)) is a variant of the Pure Random Search which includes local mutations. It starts with a random population and evolves these points by an heuristic rule.

- **DIRECT** (Dividing Rectangles, Jones et al. (1993)) is a partitioning algorithm. It uses a deterministic splitting technique of the search space and a set of admissible Lipschitz constants in order to sequentially divide and evaluate the function over a subdivision which has a chance to contain the optimum.

- **IHR** (Improving Hit-and-Run, Zabinsky et al. (1993)) is an extension of the standard Hit-and-run algorithm (Boneh and Golan (1979)) which only allows moves towards improving directions. At each step, the next direction is sampled uniformly over the unit sphere and the step size is sampled uniformly among a set of admissible values.

- **MLSL** (Multi-Level Single-Linkage, Kan and Timmer (1987)) is a multistart algorithm. It performs a series of local optimizations starting from points randomly chosen by a clustering heuristic that helps to avoid repeated searches of the same local optima.

- **PRS** (Pure Random Search, Brooks (1958)) is the standard random covering method which consists in evaluating the function over a sequence of points uniformly and independently sampled over the input space.

The source of the implementations of the algorithms can be found in Table 4.1. For a fair comparison, the tuning parameters of the algorithms were all set to default and AdaLIPO and AdaRankOpt were constantly used with a parameter p set to 0.1 and with the sequence of Lipschitz constants $k_i = (1 + 0.01/d)^i$ and the sequence of polynomial ranking rules. We stress that these parameters were chosen by an arbitrary rule of thumb.

Source	**Algorithms**
Figures 2.4 and 3.4 in Chapters 2 and 3	AdaLIPO, AdaRank
BayesOpt Library (Martinez-Cantin (2014))	BayesOpt
The CMA 1.1.06 package (Hansen (2011))	CMA-ES
NLOpt Library (Johnson (2014))	CRS, DIRECT, MLSL
Direct implementation from Zabinsky (2003)	IHR, PRS

Table 4.1: Source of the implementations of the algorithms of the benchmark

4.2.2 Test problems

The test bed includes two series of nonconvex optimization problems which involve real data sets and naturally arise in the tuning of machine learning algorithms and two series

of artificial problems that are commonly met in standard global optimization benchmarks.

Ridge regression regularization. First, we studied the task of estimating the regularization parameter $\lambda^\star$ and the bandwidth $\sigma^\star$ of a gaussian kernel ridge regression which minimize the empirical mean squared error of the predictions over a 10-fold cross validation with real data sets. The optimization was performed over $(\ln(\lambda),\ln(\sigma)) \in [-2,4] \times [-5,5]$ and five data sets taken from the UCI Machine Learning Repository (Lichman (2013)) were employed: *Auto-MPG*, *Breast Cancer Wisconsin (Prognostic)*, *Concrete slump test*, *Housing* and *Yacht Hydrodynamics*. For each data set $(x_1,y_1),\dots,(x_n,y_n)$, we considered the following optimization problem:

$$(\ln(\sigma^\star),\ln(\lambda^\star)) \in \underset{(\ln(\sigma),\ln(\lambda))\in\mathcal{X}}{\arg\min} \frac{1}{10}\sum_{k=1}^{10}\sum_{i\in I_k}(\hat{g}^k_{\sigma,\lambda}(x_i)-y_i)^2$$

where

$$\hat{g}^k_{\sigma,\lambda} \in \underset{g\in\mathcal{H}_\sigma}{\arg\min}\sum_{i\notin I_k}(g(x_i)-y_i)^2+\lambda\|g\|_{\mathcal{H}_\sigma},$$

over $\mathcal{X} = [-2,4]\times[-5,5]$ and where $I_1,\dots,I_{10}$ are 10 disjoint subsets of integers of similar sizes forming a partition of $\{1,\dots,n\}$, $\mathcal{H}_\sigma$ is the guaussian RKHS of bandwidth σ and $\|\cdot\|_{\mathcal{H}_\sigma}$ denotes the corresponding norm.

Neural nets hyperparameter calibration. Then, we studied the task of estimating the learning rate $\eta^\star$ and the momentum $\mu^\star$ of an accelerated stochastic gradient descent which minimize the empirical classification error of the predictions of a neural net with fixed architecture tuned with the stochastic gradient descent over a 5-fold cross validation involving real data sets. The optimization was performed over $(\ln(\eta),\ln(1-\mu)) \in [-5,1/2]\times[-5,0]$ and five data sets taken from the UCI Machine Learning Repository (Lichman (2013)) were employed: *Abalone*, *Indian Liver Patient*, *Prima Indians Diabetes*, *Planning Relax* and *Wine*. For each data set $(x_1,y_1),\dots,(x_n,y_n)$ with $x_i\in\mathbb{R}^d$ and $y_i\in\{0,1\}$, we considered the following optimization problem:

$$(\ln(\eta^\star),\ln(1-\mu^\star)) \in \underset{(\ln(\eta),\ln(1-\mu))\in\mathcal{X}}{\arg\min}\frac{1}{5}\sum_{k=1}^{5}\frac{1}{|I_k|}\sum_{i\in I_k}\mathbb{I}\Big\{\hat{g}^k_{\eta,\mu}(x_i)\neq y_i\Big\}$$

where

$$\hat{g}^k_{\eta,\mu}(x) = \mathbb{I}\Big\{\sigma_3(W_3^k\sigma_2(W_2^k\sigma_1(W_1^k x)))\geq 1/2\Big\}$$

over $\mathcal{X} = [-5,1/2]\times[-5,0]$ and where $I_1,\dots,I_5$ are 5 disjoint subsets of integers of similar sizes forming a partition of $\{1,\dots,n\}$, the matrix $W_1^k\in\mathbb{R}^{d\times 12}$, $W_2^k\in\mathbb{R}^{12\times 8}$ and $W_3^k\in\mathbb{R}^{8\times 1}$ are learned with an accelerated stochastic gradient descent minimizing the cross-entropy binary loss tuned with the parameters (η,μ) after 80 epochs with a batch size of 64 over the data set $\{(x_i,y_i)\}_{i\notin I_k}$ and where σ_1,σ_2 and σ_3 are respectively the ReLu, ReLu and sigmoid activation functions.

Synthetic functions I. We then compared the algorithms on a series of five two-dimensional problems taken from Jamil and Yang (2013) and Surjanovic and Bingham (2013): *Branin-Hoo*, *Himmelblau*, *McCormick*, *Levy N.13* and *Styblinski*. The dimensionality

of these problems allows an easy visualization and it can be seen that this series covers a wide variety of situations, including multimodal and non-linear functions as well as ill-conditioned and well-shaped functions. A description of the test functions can be found in table 4.2.

Problem	Objective function	Domain	Local max.
Branin-Hoo	$10(1 - 1/(8\pi))\cos(x_1) + 10 + (x_2 - 5.1x_1^2/(4\pi^2) + 5x_1/\pi - 6)^2$	$[-5,10] \times [0,15]$	3
Himmelblau	$-(x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2$	$[-5,5] \times [-5,5]$	4
Levy N.13	$-(x_1-1)^2(1+\sin^2(3\pi x_2)) -\sin^3(3\pi x_1) - (x_2-1)(1+\sin^2(2\pi x_2))$	$[-10,10] \times [-10,10]$	> 100
McCormick	$-\sin(x_1 + x_2) - (x_1 - x_2)^2 + 1.5x_1 - 2.5x_2 - 1$	$[-1.5,4] \times [-3,4]$	2
Styblinski	$8x_2^2 - 0.5x_2^4 - 2.5x_2 + 8x_1^2 - 0.5x_1^4 - 2.5x_1$	$[-5,5] \times [-5,5]$	4
Deb N.1	$\frac{1}{5}\sum_{i=1}^{5}\sin^6(5\pi x_i)$	$[-5,5]^5$	36
Holder Table	$\lvert\sin(x_1)\rvert \times \lvert\cos(x_2)\rvert \times\exp(\lvert 1-(x_1^2 + x_2^2)^{1/2}/\pi\rvert)$	$[-10,10]^2$	36
Linear Slope	$\sum_{i=1}^{4} 10^{(i-1)/4}(x_i - 5)$	$[-5,5]^7$	1
Rosenbrock	$-\sum_{i=1}^{2}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-2.048,2.048]^3$	-
Sphere	$-\left(\sum_{i=1}^{4}(x_i - \pi/16)^2\right)^{1/2}$	$[0,1]^4$	1

Table 4.2: Description of the synthetic functions of the benchmark. The top of the table displays the functions of the first series of problems while the bottom of the table displays the second series of problems. Dash symbols are used when a value could not be calculated.

Synthetic functions II. The last benchmark we used to assess the performance of the algorithms consists of a set of five synthetic functions with a dimensionality varying from two to seven. It contains the *Holder Table*, *Rosenbrock*, *Sphere*, *Linear slope* and *Deb N.1* functions taken from Finck et al. (2010) and Jamil and Yang (2013). A complete description of these functions can be found in Table 4.2. Remark that, due to the dimensionality of the input spaces, very little is known on the structure of these functions.

4.2.3 Protocol and performance metrics

For each algorithm A and objective function f, we performed $K=100$ distinct runs with a budget of $n=1000$ function evaluations. For each target parameter $t = 90\%$, 95% and 99%, we have collected the stopping times corresponding to the number of evaluations required by each method to reach the specified target

$$\tau(A,f,k,t) := \min\{i = 1,\dots,n:\ f(X_i^{(k)}) \geq f_{\text{target}}(t)\}$$

where $f(X_1^{(k)}),\dots,f(X_n^{(k)})$ denotes a sequence of n evaluations made by the algorithm A over the function f on the k-th run with $k \leq K$ and with a target value set to:

$$f_{\text{target}}(t) := \max_{x\in\mathcal{X}} f(x) - \left(\max_{x\in\mathcal{X}} f(x) - \int_{x\in\mathcal{X}} f(x)\ \mathrm{d}x/\mu(\mathcal{X})\right) \times (1-t).$$

The normalization of the target with regards to the average value of the function over the domain prevents the performance measures from being dependent of any constant term in the objective function. In practice, the average value was estimated from a Monte Carlo sampling of 10^6 evaluations and the maximum of the function was estimated, for the real task problems, by taking the best value observed over all the sets of experiments. Based on these stopping times, we measured performance through a collection of indicators:

I) Proportion of runs for which the algorithm A reached the target t over the objective function f with less than the total budget of $n = 1000$ evaluations:

$$\hat{p}_K(A,f,t) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\{\tau(A,f,k,t) < n\}.$$

II) Average and standard deviation of the number of evaluations required by the algorithm A to reach the target t over the objective function f within the total budget:

$$\overline{\tau}_K(A,f,t) = \frac{1}{|I|}\sum_{k\in I}\tau(A,f,k,t) \quad \text{and} \quad \widehat{\sigma}_\tau(A,f,t) = \left(\frac{1}{|I|}\sum_{k\in I}(\tau_k - \overline{\tau}_K)^2\right)^{1/2}$$

where $I := \{k = 1,\dots,K : \tau(A,f,k,t) < n\}$ denotes the set of the indexes of the runs that reached the target t within the total budget.

III) Aggregation of the proportion of runs for which an algorithm A reached the target t in terms of number $i \leq n$ of function evaluations over a series $f_{1:N} = \{f_1,\dots,f_N\}$ of $N = 5$ optimization problems:

$$\forall i \leq n,\ \ \widehat{\mathbb{P}}_K(A,f_{1:N},t,i) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}\{\tau(A,f_j,k,t) \leq i\}.$$

IV) Aggregation of the proportion of runs for which an algorithm A_1 has executed less (or more) evaluations to reach the target t than an algorithm A_2 over a series $f_{1:N} = \{f_1,\dots,f_N\}$ of $N = 5$ optimization problems:

$$W(A_1,A_2,f_{1:5},t) := \frac{1}{N}\sum_{j=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\big[\mathbb{I}\big\{1.1\times\tau(A_1,f_j,k,t) < \tau(A_2,f_j,k,t)\big\} - \mathbb{I}\big\{1.1\times\tau(A_2,f_j,k,t) < \tau(A_1,f_j,k,t)\big\}\big].$$

Note that this indicator considers that two algorithms performed equally when they displayed the same results within a 10% threshold. Remark also that if $W = 1$, then A_1 always performed better than A_2, if $W = 0$ they performed equally and if $W = -1$ then A_1 was always worst than A_2.

Indicators (I) and (II) record the performance of an algorithm over a single objective function while indicators (III) and (IV) aggregate its performance over each series of five optimization problems of the benchmark (*i.e.*, ridge regression regularization, neural nets hyperparameter tuning, Synthetic functions I and II). These indicators capture important properties of global optimization algorithms, such as accuracy, stability and speed of convergence.

4.3 Results and examples

Preliminaries. Before collecting the results, we first present a simple experiment that illustrates the different strategies behind each solver. For an easy visualization, we considered the problem of maximizing the two-dimensional Styblinski function:

$$f(x_1, x_2) = -\frac{1}{2}\sum_{i=1}^{2}(x_i^4 - 16x_i^2 + 5x_i)$$

over the domain $\mathcal{X} = [-5,5] \times [-5,5]$. Each solver was run with a limit of $n = 50$ function evaluations. The sequences of evaluation points generated by each methods are displayed on Figure 4.3. Several remarks should be made:

- DIRECT is the only deterministic algorithm of the benchmark. We also point out that it is a function value-free solver (*i.e.,* for any strictly increasing function $g : \mathbb{R} \to \mathbb{R}$ and any target function $f : \mathcal{X} \to \mathbb{R}$, it generates the same points over f and $g \circ f$).

- MLSL is the only algorithm of the benchmark that performs local searches during the optimization process. Remark also that, similarly to BayesOpt, it performs less exploration than the other methods by default.

- The CRS and CMA-ES algorithms start by exploring the space with few evaluations before actually optimizing the function. Recall however that the number of evaluations dedicated to the exploration phase can be tuned by an hyperparameter.

- Lastly, we point out that both the BayesOpt and IHR algorithms can get stuck in a corner if it contains a local optima. In the case of IHR, it comes from the uniform generation of the direction, but we point out that several extensions allowing arbitrary distributions have been proposed in Bélisle et al. (1993).

Keeping these observations in mind, we may now present the results of the experiments.

Results. Results are collected in a series of figures and tables. The proportion of runs that reached the specified targets in terms of function evaluations (Indicator III) are displayed in Figure 4.2. The numbers of runs for which an algorithm has executed less evaluation than another algorithm (Indicator IV) are displayed in Figure 4.3. Finally, the number

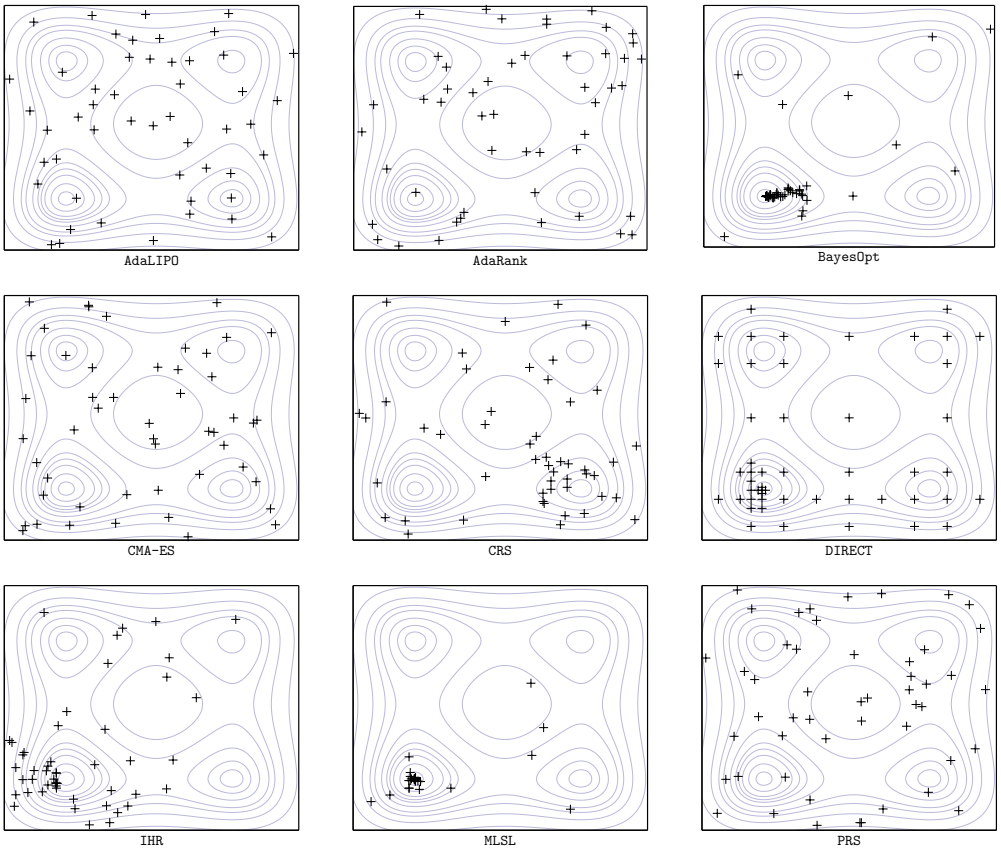


Figure 4.1: Series of $n = 50$ evaluations points generated by each algorithm on the two-dimensional Styblinski function. The input space $\mathcal{X}$ is set to $[-5,5] \times [-5,5]$ and the optimum of the function is located on the bottom left corner.

of runs that reached the target within the total budget, as well as the average number and standard deviation of iterations required to reach the target (indicators I and II) are collected in Tables 4.4 and 4.5.

4.4 Discussion and perspectives

In this section, we discuss the results of the numerical experiments and propose several extensions that might improve the performance of AdaLIPO and AdaRank.

General remarks. We start with a general discussion on the experiments:

- As one should expect, there is no algorithm that performs uniformly better than the other over each problem of the benchmark. Note that this makes the comparison difficult as one can always find a problem for which an algorithm is better than another.

- On the contrary, except on the neural nets tuning problems, the Pure Random Search (PRS) tends to be outperformed by any method over most problems of the benchmark (see, e.g., the proportion of runs that reached a near-optimal solution in terms

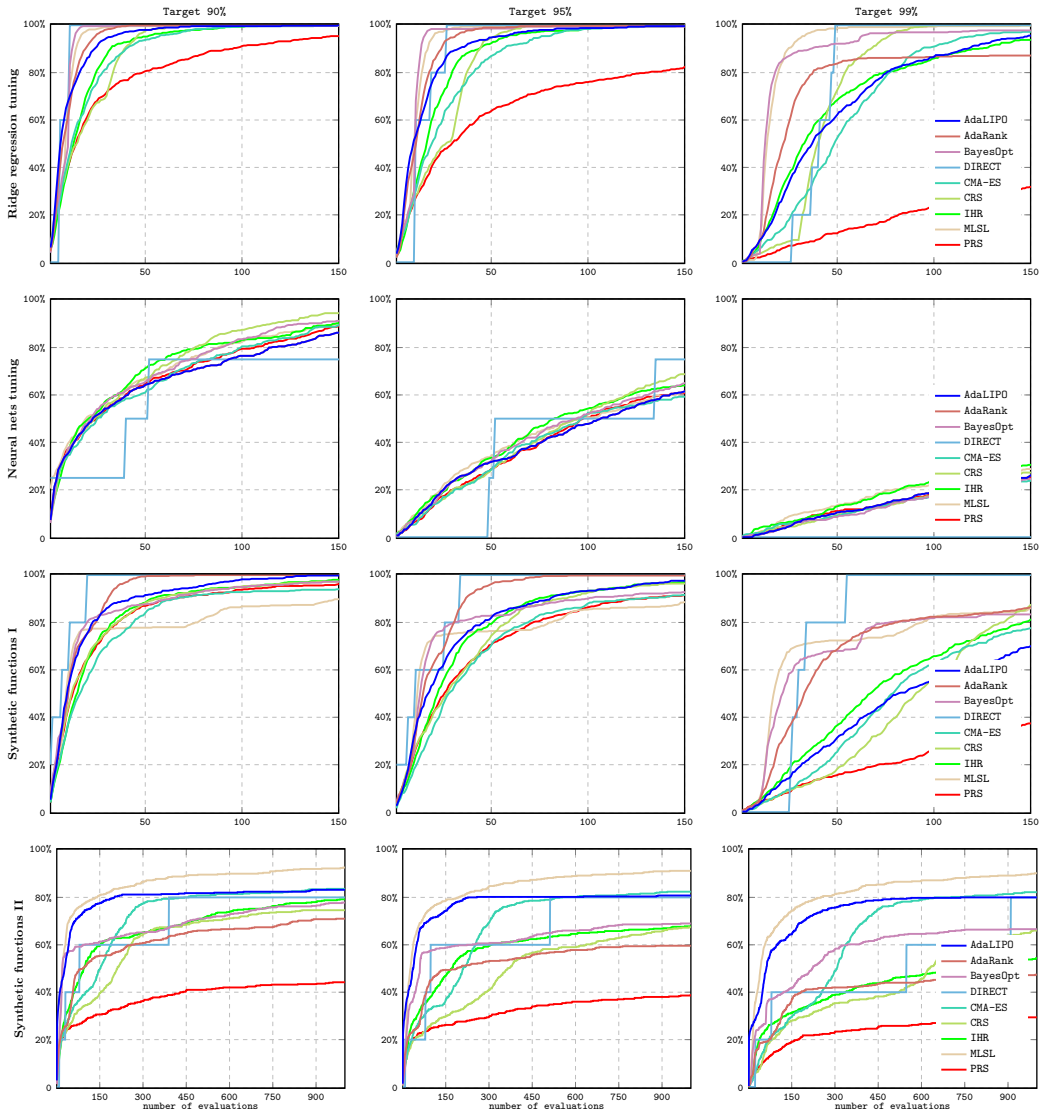


Figure 4.2: Aggregation of the proportion of runs that reached the targets 90%, 95% and 99% in terms of function evaluations over each series of optimization problems.

of function evaluations in Figure 4.2). This simple observation suggests that using a global optimization algorithm generally lead to better results than the naive random search over a large class of problems.

- Before turning to a finer analysis, it is important to note that as DIRECT is a deterministic method and always starts with the same initial points, it can sometimes find a near-global solution with one iteration regardless of the shape of the function (*e.g.* when the maximum is located on the center of the domain, like the Levy N.13 function). Since this has an impact on the indicators that aggregate its performance (e.g. Figures 4.3 and 4.2), we stress that the results of DIRECT should be read carefully.

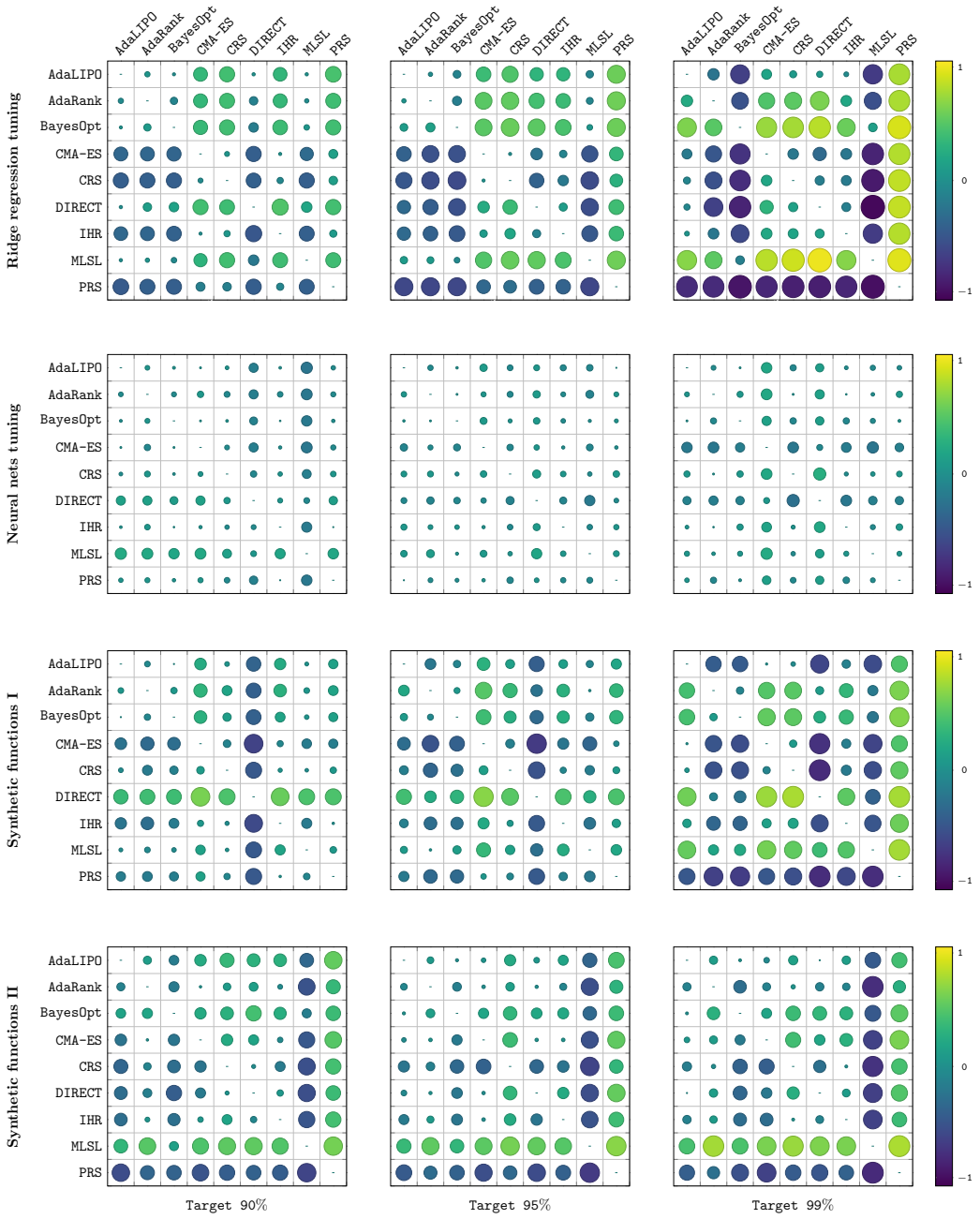


Figure 4.3: Aggregation of the proportion of runs for which an algorithm has executed less (or more) evaluations to reach a specified target than another algorithm over each series of optimization problems. The matrix display the indicators $W(A_1, A_2, \cdot, \cdot)$ described in Section 4.2 where A_1 is located on the raw and A_2 on the column.

- In terms of the proportion of runs that reached a near-optimal solution within the given budget, it should be noticed that most methods are able to reach the 95% over most problems of the benchmark (see Tables 4.5 and 4.4) while there is no method that is able to solve more than 90% of the neural nets tuning problems with a target

Problem	AdaLIPO		AdaRank		BayesOpt		CMA-ES		CRS		DIRECT		IHR		MLSL		PRS	
Ridge regression regularization, Target 90%																		
Auto	100%	14.6(008)	100%	13.7(005)	100%	**10.8**(002)	100%	29.3(024)	100%	28.7(014)	100%	11.0(000)	100%	24.9(020)	100%	13.1(015)	100%	65.1(062)
Breast	100%	**05.4**(003)	100%	06.1(003)	100%	06.8(003)	100%	11.1(008)	100%	08.9(007)	100%	06.0(000)	100%	11.4(008)	100%	06.6(003)	100%	10.6(009)
Conc.	100%	**04.9**(002)	100%	05.8(003)	100%	06.4(003)	100%	10.4(008)	100%	10.0(008)	100%	06.0(000)	100%	10.4(007)	100%	06.1(003)	100%	09.8(009)
Hous.	100%	**05.5**(003)	100%	06.5(003)	100%	07.5(003)	100%	12.4(012)	100%	13.8(010)	100%	06.0(000)	100%	12.8(009)	100%	07.2(003)	100%	11.5(009)
Yacht	100%	25.2(021)	100%	17.3(008)	100%	13.8(020)	100%	29.6(025)	100%	32.6(014)	100%	**11.0**(000)	100%	27.6(022)	100%	14.4(013)	100%	73.3(072)
Total	100%	11.1(007)	100%	09.9(004)	100%	09.1(006)	100%	18.6(015)	100%	18.8(011)	100%	**08.0**(000)	100%	17.4(013)	100%	09.5(007)	100%	34.1(032)
Ridge regression regularization, Target 95%																		
Auto	100%	17.7(009)	100%	19.4(031)	100%	12.2(005)	100%	42.9(031)	100%	35.8(012)	100%	**11.0**(000)	100%	29.6(022)	100%	15.0(015)	100%	139 (131)
Breast	100%	**06.6**(004)	100%	06.9(003)	100%	08.3(003)	100%	13.7(010)	100%	13.6(010)	100%	11.0(000)	100%	13.1(009)	100%	07.6(003)	100%	17.6(017)
Conc.	100%	**06.4**(003)	100%	06.7(003)	100%	07.9(003)	100%	13.4(010)	100%	14.6(010)	100%	11.0(000)	100%	12.0(007)	100%	07.3(003)	100%	14.0(012)
Hous.	100%	17.9(025)	100%	11.7(004)	100%	13.9(022)	100%	23.1(016)	100%	22.8(012)	100%	19.0(000)	100%	20.2(013)	100%	**11.7**(007)	100%	39.6(039)
Yacht	100%	33.3(026)	100%	23.4(012)	100%	**15.9**(021)	100%	40.5(030)	100%	38.3(014)	100%	27.0(000)	100%	33.6(026)	100%	16.2(013)	98%	232 (227)
Total	100%	16.4(013)	100%	13.6(011)	100%	11.7(011)	100%	26.7(019)	100%	25.0(012)	100%	15.8(000)	100%	21.7(015)	100%	**11.6**(008)	99%	87.8(085)
Ridge regression regularization, Target 99%																		
Auto	100%	32.6(016)	96%	56.4(142)	100%	**14.0**(007)	100%	73.7(049)	100%	48.5(016)	100%	47.0(000)	100%	64.3(070)	100%	20.6(017)	47%	462 (279)
Breast	100%	34.1(036)	100%	16.0(010)	100%	31.1(051)	100%	35.1(020)	100%	34.8(012)	100%	27.0(000)	100%	32.8(035)	100%	**12.8**(003)	100%	146 (124)
Conc.	100%	70.8(058)	100%	22.1(011)	100%	28.1(033)	100%	46.3(029)	100%	36.5(014)	100%	37.0(000)	100%	45.3(044)	100%	**14.7**(010)	100%	177 (148)
Hous.	100%	65.4(062)	100%	22.5(010)	100%	17.9(022)	100%	61.5(085)	100%	43.7(014)	100%	41.0(000)	100%	51.9(049)	100%	**16.3**(010)	92%	355 (270)
Yacht	100%	61.7(039)	65%	151 (208)	100%	**18.5**(022)	100%	70.9(049)	100%	52.9(018)	100%	49.0(000)	100%	70.1(068)	100%	21.4(014)	39%	433 (301)
Total	100%	53.0(042)	92%	46.2(065)	100%	21.9(027)	100%	57.5(047)	100%	43.3(015)	100%	40.2(000)	100%	52.9(053)	100%	**17.2**(011)	75%	46 (203)
Neural nets hyperparameter tuning, Target 90%																		
Aba	100%	144 (139)	99%	136 (128)	100%	100 (098)	100%	109 (118)	100%	**78.6**(080)	100%	164 (000)	100%	108 (108)	100%	104 (093)	100%	120 (103)
Il	100%	**23.1**(023)	100%	38.3(037)	100%	31.1(029)	96%	31.5(029)	100%	34.2(032)	100%	40.0(000)	100%	24.1(023)	100%	35.6(049)	100%	27.6(026)
Plrx	100%	03.2(002)	100%	03.4(002)	100%	03.4(002)	100%	03.9(002)	100%	04.2(003)	100%	**01.0**(000)	100%	04.8(004)	100%	02.1(005)	100%	03.4(002)
Prima	100%	111 (096)	100%	70.7(065)	100%	73.6(069)	100%	101 (114)	100%	66.3(058)	100%	**52.0**(000)	99%	71.7(071)	100%	68.8(074)	100%	84.7(078)
Wine	100%	03.1(002)	100%	03.9(003)	100%	04.1(003)	100%	04.0(004)	100%	03.8(003)	100%	**01.0**(000)	100%	04.8(004)	100%	03.3(003)	100%	03.8(003)
Total	100%	57.0(052)	99%	50.4(047)	100%	42.5(040)	99%	50.1(053)	100%	**37.4**(035)	100%	51.6(000)	99%	42.7(042)	100%	42.7(045)	100%	48.0(042)
Neural nets hyperparameter tuning, Target 95%																		
Aba	85%	398 (270)	92%	373 (254)	90%	314 (254)	84%	324 (264)	95%	**238** (219)	100%	701 (000)	91%	298 (237)	89%	317 (254)	88%	363 (267)
Il	100%	63.5(060)	100%	76.4(072)	100%	60.1(057)	90%	70.4(061)	100%	75.9(057)	100%	135 (000)	100%	**53.6**(053)	100%	69.7(075)	100%	79.5(085)
Plrx	100%	89.5(084)	100%	118 (112)	100%	89.6(078)	75%	52.0(039)	99%	106 (101)	100%	**49.0**(000)	100%	89.2(078)	100%	84.8(086)	100%	94.1(094)
Prima	100%	221 (198)	100%	137 (133)	100%	165 (162)	96%	192 (193)	100%	144 (135)	100%	**52.0**(000)	98%	178 (172)	98%	176 (159)	100%	182 (160)
Wine	100%	03.9(002)	100%	04.6(004)	100%	04.6(003)	100%	05.9(008)	100%	04.7(004)	100%	**03.0**(000)	100%	08.4(012)	100%	06.0(012)	100%	04.6(003)
Total	97%	148 (118)	98%	138 (113)	98%	123 (108)	89%	127 (112)	98%	**112** (102)	100%	188 (000)	97%	122 (108)	97%	126 (114)	97%	139 (118)
Neural nets hyperparameter tuning, Target 99%																		
Aba	68%	413 (270)	72%	446 (279)	69%	364 (257)	53%	401 (264)	72%	**302** (239)	100%	701 (000)	66%	368 (268)	68%	366 (273)	70%	410 (273)
Il	98%	229 (206)	100%	217 (198)	98%	241 (204)	79%	**156** (136)	98%	232 (161)	100%	471 (000)	100%	164 (152)	100%	224 (203)	99%	259 (219)
Plrx	75%	370 (264)	76%	375 (272)	71%	422 (271)	29%	177 (172)	82%	387 (260)	0%	**00.0**(000)	75%	355 (246)	75%	357 (282)	68%	341 (262)
Prima	76%	356 (278)	91%	301 (226)	85%	302 (223)	69%	333 (265)	88%	266 (220)	100%	**209** (000)	85%	339 (267)	85%	355 (261)	88%	310 (228)
Wine	100%	07.9(007)	100%	08.8(008)	100%	07.6(005)	100%	12.2(013)	100%	09.9(009)	100%	**03.0**(000)	100%	13.0(018)	100%	11.0(018)	100%	09.7(009)
Total	83%	254 (192)	87%	252 (187)	84%	249 (180)	66%	**190** (149)	88%	229 (169)	80%	346 (000)	85%	229 (178)	85%	246 (196)	85%	249 (187)

Figure 4.4: Results of the numerical experiments on the ridge regression and neural nets tuning problems. The table display the proportion of runs that reached the specified targets and the average and standard deviation of evaluations required to reach the target (format: %, mean, (standard deviation)). In bold, the best result obtained in terms of average of function evaluations.

set to 99%.

- In terms of the total number of evaluations required to reach a near-optimal solution, the MLSL algorithm obtains several times the best results over the Ridge regression regularization problems and the second series of synthetic functions (see Tables 4.5 and 4.4) while the CRS and DIRECT algorithms respectively obtains several times the best results over the neural nets tuning problems and the first series of synthetic functions.

- In terms of speed of convergence, the BayesOpt and MLSL algorithms need in average less function evaluations to reach a near-optimal solution when they converge than the CMA-ES, CRS and IHR algorithms (see, e.g., Figure 4.3 or Table 4.4). Moreover, as seen on Figure 4.3, the AdaLIPO and AdaRank algorithms generally rank between the latter.

We may now discuss the performance of the methods developed in the thesis.

AdaLIPO. Here we focus on the performance of AdaLIPO. Our main observations are the following:

Problem	AdaLIPO		AdaRank		BayesOpt		CMA-ES		CRS		DIRECT		IHR		MLSL		PRS	
Synthetic functions I, Target 90%																		
Branin	100%	08.4(005)	100%	07.3(004)	100%	**07.1**(004)	100%	22.0(018)	100%	09.2(008)	100%	11.0(000)	100%	17.4(019)	100%	09.2(007)	100%	11.1(008)
Himm.	100%	14.5(010)	100%	12.2(008)	100%	12.7(013)	100%	18.0(014)	100%	13.4(013)	100%	**02.0**(000)	100%	15.9(012)	100%	07.6(004)	100%	14.4(012)
Levy	100%	11.2(007)	100%	13.1(012)	100%	10.4(006)	100%	17.0(014)	100%	17.9(016)	100%	**01.0**(000)	100%	18.1(012)	98%	15.9(070)	100%	19.9(018)
McCo.	100%	09.0(007)	100%	09.8(007)	100%	07.8(005)	98%	13.1(015)	100%	14.7(011)	100%	**07.0**(000)	100%	14.2(014)	98%	56.2(102)	100%	15.4(016)
Stybl.	100%	48.9(040)	100%	27.0(011)	100%	79.9(079)	79%	62.6(055)	89%	56.5(039)	100%	**20.0**(000)	100%	63.7(061)	100%	116 (090)	100%	82.7(078)
Total	100%	18.4(014)	100%	13.9(008)	100%	23.6(021)	95%	25.0(022)	97%	21.6(017)	100%	**08.2**(000)	100%	25.9(024)	99%	41.1(054)	100%	28.7(027)
Synthetic functions I, Target 95%																		
Branin	100%	14.0(010)	100%	10.8(005)	100%	**10.6**(004)	100%	31.1(021)	100%	22.5(019)	100%	11.0(000)	100%	25.7(023)	100%	19.7(035)	100%	21.3(021)
Himm.	100%	29.1(024)	100%	18.9(010)	100%	20.8(020)	100%	38.2(027)	100%	31.3(029)	100%	26.0(000)	100%	26.9(022)	100%	**10.1**(004)	100%	31.2(034)
Levy	100%	19.9(016)	100%	19.7(022)	100%	14.6(006)	100%	27.0(023)	100%	27.6(020)	100%	**01.0**(000)	100%	26.9(018)	98%	23.6(085)	100%	36.5(035)
McCo.	100%	16.2(012)	100%	17.4(014)	100%	10.5(006)	98%	22.6(020)	97%	25.8(018)	100%	**07.0**(000)	100%	20.2(021)	98%	60.5(103)	100%	36.1(037)
Stybl.	100%	80.6(058)	100%	**32.9**(012)	100%	150 (146)	73%	94.8(069)	89%	68.0(039)	100%	34.0(000)	100%	77.4(065)	100%	118 (089)	100%	151 (148)
Total	100%	32.0(024)	100%	19.9(013)	100%	41.4(037)	94%	39.8(030)	97%	34.3(025)	100%	**15.8**(000)	100%	35.4(030)	99%	46.4(063)	100%	55.3(055)
Synthetic functions I, Target 99%																		
Branin	100%	187 (152)	100%	39.5(106)	100%	**21.0**(010)	100%	97.8(062)	100%	101 (044)	100%	27.0(000)	100%	177 (150)	96%	41.9(065)	76%	363 (263)
Himm.	100%	102 (087)	100%	35.8(013)	100%	32.2(023)	100%	96.7(083)	100%	89.0(045)	100%	55.0(000)	100%	72.5(057)	100%	**15.2**(004)	99%	152 (166)
Levy	100%	124 (139)	100%	184 (230)	100%	37.2(028)	98%	94.8(086)	100%	90.9(039)	100%	30.0(000)	100%	73.0(052)	96%	**28.5**(066)	90%	320 (238)
McCo.	100%	47.6(033)	99%	101 (146)	100%	**16.3**(012)	98%	62.4(071)	97%	61.2(028)	100%	26.0(000)	100%	47.5(043)	98%	69.8(111)	100%	157 (132)
Stybl.	100%	224 (139)	100%	63.3(050)	63%	369 (278)	69%	168 (129)	89%	116 (045)	100%	**34.0**(000)	100%	125 (082)	100%	121 (090)	66%	362 (261)
Total	100%	137 (110)	99%	84.7(109)	92%	73.2(053)	93%	100.0(083)	97%	91.2(040)	100%	**34.4**(000)	100%	98.8(077)	98%	55.6(067)	86%	258 (205)
Synthetic functions II, Target 90%																		
DebD	16%	472 (286)	11%	538 (326)	39%	523 (239)	20%	647 (193)	4%	493 (216)	0%	00.0(000)	52%	524 (243)	87%	**78.1**(109)	6%	607 (293)
Holder.	100%	77.5(058)	100%	171 (185)	71%	169 (210)	99%	70.6(069)	73%	**51.6**(029)	100%	80.0(000)	100%	53.9(058)	84%	172 (247)	100%	190 (167)
Linear.	100%	197 (146)	100%	54.6(009)	79%	138 (230)	99%	205 (070)	96%	342 (206)	100%	390 (000)	44%	392 (222)	100%	**27.5**(036)	0%	00.0(000)
Rosen.	100%	06.8(004)	100%	**06.2**(005)	100%	07.6(005)	100%	10.0(009)	100%	09.0(008)	100%	10.0(000)	100%	12.4(009)	100%	06.9(004)	100%	08.9(009)
Sphere	100%	36.2(012)	44%	394 (272)	100%	**19.3**(002)	100%	171 (068)	100%	224 (054)	100%	31.0(000)	100%	89.7(047)	92%	104 (187)	15%	491 (272)
Total	83%	94.4(064)	71%	131 (100)	77%	118 (111)	83%	140 (061)	74%	166 (078)	80%	128 (000)	79%	152 (085)	92%	**74.0**(111)	44%	140 (106)
Synthetic functions II, Target 95%																		
DebD	4%	634 (302)	1%	**84.0**(000)	12%	569 (181)	16%	695 (153)	1%	614 (000)	0%	00.0(000)	25%	562 (235)	87%	98.4(134)	1%	746 (000)
Holder.	100%	102 (065)	94%	240 (215)	71%	180 (205)	97%	109 (105)	54%	222 (295)	100%	80.0(000)	100%	**69.7**(070)	82%	165 (234)	92%	314 (250)
Linear.	29%	584 (271)	100%	76.2(015)	62%	140 (229)	99%	273 (070)	82%	455 (265)	100%	512 (000)	14%	533 (265)	100%	**37.7**(057)	0%	00.0(000)
Rosen.	100%	11.5(010)	100%	09.3(007)	100%	11.9(008)	100%	16.1(012)	100%	15.8(014)	100%	10.0(000)	100%	15.2(011)	100%	**08.8**(005)	100%	17.0(017)
Sphere	100%	**42.1**(011)	3%	543 (236)	100%	44.8(016)	100%	224 (057)	100%	340 (066)	100%	98.0(000)	100%	173 (076)	87%	111 (166)	1%	959 (000)
Total	66%	105 (053)	59%	110 (078)	69%	98.3(096)	82%	176 (064)	67%	254 (136)	80%	175 (000)	67%	139 (074)	91%	**79.9**(113)	38%	166 (127)
Synthetic functions II, Target 99%																		
DebD	0%	00.0(000)	0%	00.0(000)	0%	00.0(000)	16%	758 (147)	0%	00.0(000)	0%	00.0(000)	4%	616 (094)	87%	**146** (182)	0%	00.0(000)
Holder.	100%	213 (129)	37%	481 (275)	71%	186 (203)	96%	182 (118)	54%	257 (287)	100%	**80.0**(000)	100%	158 (115)	82%	173 (233)	47%	449 (271)
Linear.	0%	00.0(000)	100%	128 (032)	62%	142 (229)	99%	374 (087)	76%	490 (272)	100%	910 (000)	0%	00.0(000)	100%	**50.5**(080)	0%	00.0(000)
Rosen.	100%	55.9(057)	100%	25.4(019)	100%	27.7(022)	100%	43.5(037)	100%	42.8(023)	100%	24.0(000)	100%	24.9(016)	100%	**19.4**(049)	100%	70.3(061)
Sphere	100%	**53.0**(010)	0%	00.0(000)	100%	222 (077)	100%	308 (060)	100%	607 (081)	100%	548 (000)	68%	584 (219)	82%	152 (205)	0%	00.0(000)
Total	60%	107 (065)	47%	140 (065)	66%	141 (116)	82%	248 (078)	66%	352 (141)	80%	390 (000)	54%	222 (104)	90%	**103** (143)	29%	191 (128)

Figure 4.5: Results of the numerical experiments on synthetic problems. The table display the proportion of runs that reached the specified targets and the average and standard deviation of evaluations required to reach the target (format: %, mean, (standard deviation)). In bold, the best result obtained in terms of average of function evaluations.

- First, we point out that AdaLIPO displays competitive results over most of the problems of the benchmark (see, e.g., Figure 4.2). In particular, it obtains several times the best performance for the 90% and 95% targets (see, *e.g.*, *BreastCancer* and *Housing* in Table 4.4 and 4.2) and experiment *Sphere* also suggest that, in the case of smooth functions, it can be robust against the dimensionality of the input space.

- However, in some cases where the function is not Lipschitz (*Deb.N.1*) or simply when a bound on the Lipschitz constant is not informative (*Styblinski*), it can be observed that the algorithm can either need many function evaluations to reach a near-optimal solution or never converge (see, e.g., the results in Table 4.2). A promissing approach to solve this issue would be to extend the algorithm to sets of locally smooth functions or Hölder continuous functions as in Munos (2014).

- Last, in some cases, the algorithm can be witnessed to reach the 95% target with very few evaluations while getting more slowly to the 99% target (see, *e.g.*, *Concrete, Housing*). This problem is due to the instability of the Lipschitz constant estimate when there are many local optima around the global maximum. Note that this issue could certainly be solved with the addition of a noise parameter that would allow the Lipschitz constant estimate be more robust against this type of perturbations.

AdaRank. We now focus on the performance of AdaRank. Our main observations are the following:

- The algorithm displays, as one should expect, competitive results on test problems with estimated ranking rules of moderate complexity with regards to the sequence of ranking structures set as input (see, *e.g.*, *Breast Cancer*, *Concrete*, *Housing*, *Himmelblau* or *Styblinski*). Moreover, experiment *Linear Slope* also confirm that the algorithm can be robust against the dimensionality of the input space in the case of test functions with estimated ranking rule of low complexity.

- In contrast, the method stalls on test problems which do not admit an estimated ranking rule of moderate complexity (see, *e.g.*, *Deb N.1* and *Holder Table*). Indeed, the algorithm cannot estimate efficiently the ranking rules induced by some classes of functions with a single sequence of ranking structures set as input. Considering at the same time multiple sequences of ranking structures might be a promising approach to address this issue by allowing the algorithm to adapt to a wider variety of shapes.

- Finally, in the case of test functions with strong global structure but many local optima, the algorithm reaches the 95% target with few function evaluations but fails at getting to the 99% target (see, *e.g.*, *Levy N.13*). Indeed, it starts moving toward the global optima by learning the global structure of the function but then considers ranking rules of a level of complexity higher than required when many local variations are met. Again, as detailed in Remark 3.12, adding a noise parameter would allow the algorithm to be more robust against this type of local perturbations.

Perspectives. These empirical results aim at (i) providing numerical evidence that the main algorithms of the thesis can be effective on a wide range of optimization problems and competitive with the state-of-the-art methods, and (ii) identifying some of their limits that could be solved with further extensions. Keeping in mind that a complete and detailed empirical analysis of the merits and limitations of the algorithms with these extensions is beyond the scope of the thesis, we provide as a conclusion a summary of the main extensions that could be considered in order to improve their performance:

1. Adding a noise parameter to the algorithms. As detailed above, extending the algorithms to settings with noisy evaluations could allow the algorithms to be robust against local perturbations and thus allow them to reach higher precision targets with less function evaluations by improving their stability.

2. Investigating a better exploration/exploitation trade-off. Recall indeed that the parameter p controlling this trade-off has been set to 0.1 by default for both the algorithms and regardless of the function values. As mentioned in Chapter 2, a promising approach would be to consider a dynamic sequence of parameters $p_{t\geq 1}$ that depends on both the function values and shape of the function.

3. Investigating a finer model selection. Again, since both the algorithms were run with a single sequence of Lipschitz constants and ranking structures set as input, investigating finer sequences for the model selection could certainly lead to better empirical results. As an example, a plausible strategy would be to consider at the

same time multiple sequences of ranking structures an picking among them the one that explains the function values with the ranking rule of the lowest complexity.

4. Using near-random covering methods instead of the standard Pure Random Search. Indeed, since the points generated by the Random Search are independent from one another, they do not necessarily lead to an efficient covering of the space. Thus, using more sophisticated near-random covering methods such as the on proposed in Sobol (1967) and McKay et al. (1979) could certainly improve their performance.

5. Performing local searches during the optimization process. As seen for the MLSL algorithm, incorporating local searches would allow the algorithms to focus more on promising areas before exploring the space and could thus significantly improve their performance.

6. Considering alternative sampling strategies. As detailed at the end of Chapters 2 and 3, a last approach would be to consider either more selective sampling strategies or setting a distribution on the ranking rules induced by the unknown function in order to derive a bayesian version of the algorithm.

Appendix

Abstract. This appendix contains some additional results that are used several times in the document and presents a preliminary work that has been done in the field of bayesian optimization. First, we collect some geometric inequalities which are often used in the computations. Then, we draw a short analysis of the Pure Random Search strategy and provide some convergence results that are used many times for comparison. Finally, we present a preliminary work that has been done in the field bayesian optimization, which differs from the framework considered up to this point.

Contents

A Geometric inequalities

In this section, we collect some geometric inequalities that we use repeatedly in the computations. We start by presenting a set of standard inequalities:

$$\text{For } x \in \mathbb{R}, \qquad 1 + x \leq e^x$$

$$\text{For } x \in \mathbb{R}, \qquad \lceil x \rceil \leq x + 1$$

$$\text{For } x \in \mathbb{R}, \qquad x - 1 \leq \lfloor x \rfloor$$

$$\text{For } x < 1, \qquad \ln(1 - x) \leq -x$$

$$\text{For } (x, n) \in [0,1] \times \mathbb{N} \qquad 1 - x \leq \left(1 - \tfrac{x}{n}\right)^n$$

$$\text{For } x \geq 0, y \geq 0 \qquad \sqrt{x+y} \leq \sqrt{x} + \sqrt{y}.$$

We may now provide three geometric results (Corollary A.3, Lemma A.4 and Lemma A.5) that are used several times in the proofs. We start with the definition of covering numbers.

Definition A.1. (Covering number and ϵ-cover) *For any compact and convex set $\mathcal{X} \subset \mathbb{R}^d$ and any $\epsilon > 0$, we say that a sequence $x_1, \ldots, x_n$ of n points in $\mathcal{X}$ defines an ϵ-cover of $\mathcal{X}$ if and only if $\mathcal{X} \subseteq \bigcup_{i=1}^n B(x_i, \epsilon)$. The covering number $\mathcal{N}_\epsilon(\mathcal{X})$ of $\mathcal{X}$ is then defined as the minimal size of a sequence defining an ϵ-cover of $\mathcal{X}$, i.e.*

$$\mathcal{N}_\epsilon(\mathcal{X}) := \inf \left\{ n \in \mathbb{N}^\star : \exists (x_1, \ldots, x_n) \in \mathcal{X}^n \text{ s.t. } \mathcal{X} \subseteq \bigcup_{i=1}^n B(x_i, \epsilon) \right\}.$$

The next result provides an upper bound on the covering numbers of hypercubes.

Proposition A.2. (Covering number of hypercubes) *Let $[0,R]^d$ be an hypercube of dimensionality $d \geq 1$ whose side has length $R > 0$. Then, for all $\epsilon > 0$, we have that*

$$\mathcal{N}_\epsilon([0,R]^d) \leq (\sqrt{d}R/2\epsilon)^d \vee 1.$$

Proof. Observe first that since $[0,R]^d \subseteq B(c, \sqrt{d}R/2)$ where c denotes the center of the hypercube, the result trivially holds for any $\epsilon \geq \sqrt{d}R/2$. Now, fix any $\epsilon < \sqrt{d}R/2$, set $N_\epsilon = \lceil \sqrt{d}R/2\epsilon \rceil$ and define for all $I \in \{0, \ldots, N_\epsilon - 1\}^d$ the series $H_I = I \times R/N_\epsilon + [0, R/N_\epsilon]^d$ of N_ϵ^d hypercubes which cover $[0,R]^d := \bigcup_{I \in \{0,\ldots,N_\epsilon-1\}^d} H_I$. Denoting by c_I the center of H_I and observing $\max_{x \in H_I} \|x - c_I\|_2 \leq \epsilon$, we directly deduce that $H_I \subseteq B(c_I, \epsilon)$ which implies that $[0,R]^d \subseteq \bigcup_{I \in \{0,\ldots,N_\epsilon-1\}^d} B(c_I, \epsilon)$ and proves that $\mathcal{N}_\epsilon([0,R]^d) \leq N_\epsilon^d \leq (\sqrt{d}R/2\epsilon)^d$. □

This result can be extended to any compact and convex set of $\mathbb{R}^d$ as shown below.

Corollary A.3. (Covering number of a convex set) *For any bounded compact and convex set $\mathcal{X} \subset \mathbb{R}^d$, we have that $\forall \epsilon > 0$,*

$$\mathcal{N}_\epsilon(\mathcal{X}) \leq (\sqrt{d}\,\mathrm{diam}(\mathcal{X})/\epsilon)^d \vee 1.$$

Proof. First, we show that $\mathcal{N}_\epsilon(\mathcal{X}) \leq \mathcal{N}_\epsilon([0,2\,\mathrm{diam}(\mathcal{X})]^d)$ and second, we use the bound of Proposition A.2 to conclude the proof. By definition of $\mathrm{diam}(\mathcal{X})$, we know that there exists some $x \in \mathbb{R}^d$ such that $\mathcal{X} \subseteq x + [0,2\,\mathrm{diam}(\mathcal{X})]^d$. Hence, it follows from Proposition A.2 that there exists a sequence $c_1,\ldots,c_{N_\epsilon}$ of $N_\epsilon \leq \mathcal{N}_\epsilon([0,2\,\mathrm{diam}(\mathcal{X})]^d)$ points in $[0,2\,\mathrm{diam}(\mathcal{X})]^d$ forming an ϵ-cover of $\mathcal{X}$:

$$\mathcal{X} \subseteq [0,2\,\mathrm{diam}(\mathcal{X})]^d \subseteq \bigcup_{i=1}^{N_\epsilon} B(c_i,\epsilon). \tag{A.1}$$

However, we do not have the guarantee at this point that the centers $c_1,\ldots c_{N_\epsilon}$ belong to $\mathcal{X}$. To build an ϵ-cover of $\mathcal{X}$, we simply project each of the centers on $\mathcal{X}$. More precisely, we show that $\mathcal{X} \subseteq \bigcup_{i=1}^{N_\epsilon} B(\Pi_\mathcal{X}(c_i),\epsilon)$ where $\Pi_\mathcal{X} : x \in \mathbb{R}^d \mapsto \arg\min_{x' \in \mathcal{X}} \|x - x'\|_2 \in \mathcal{X}$ denotes the projection over the compact and convex set $\mathcal{X}$. Starting from (A.1), it sufficient to show that $B(c_i,\epsilon) \cap \mathcal{X} \subseteq B(\Pi_\mathcal{X}(c_i),\epsilon)$, for all $i \in \{1,\ldots,N_\epsilon\}$ to prove that

$$\mathcal{X} \subseteq \bigcup_{i=1}^{N_\epsilon} B(c_i,\epsilon) \cap \mathcal{X} \subseteq \bigcup_{i=1}^{N_\epsilon} B(\Pi_\mathcal{X}(c_i),\epsilon).$$

Pick any $c \in \{c_1,\ldots,c_{N_\epsilon}\}$ and consider the following cases on the distance $\|c - \Pi_\mathcal{X}(c)\|_2$ between the center and its projection: (i) if $\|c - \Pi_\mathcal{X}(c)\|_2 = 0$, then $c = \Pi_\mathcal{X}(c)$ and we have $B(c,\epsilon) \cap \mathcal{X} \subseteq B(\Pi_\mathcal{X}(c),\epsilon)$, (ii) If $\|c - \Pi_\mathcal{X}(c)\|_2 > \epsilon$, then $\mathcal{X} \cap B(c,\epsilon) = \emptyset$, and we have $\mathcal{X} \cap B(c,\epsilon) \subseteq B(\Pi_\mathcal{X}(c),\epsilon)$. We now consider the non-trivial case where $\|c - \Pi_\mathcal{X}(c)\|_2 \in (0,\epsilon)$. Pick any $x \in B(c,\epsilon) \cap \mathcal{X}$ and note that as $x \in B(c,\epsilon)$, then

$$\begin{aligned}
\epsilon^2 &\geq \|x - c\|_2^2 \\
&= \|x - \Pi_\mathcal{X}(c) + \Pi_\mathcal{X}(c) - c\|_2^2 \\
&= \|x - \Pi_\mathcal{X}(c)\|_2^2 + \|c - \Pi_\mathcal{X}(c)\|_2^2 + 2 \cdot \langle x - \Pi_\mathcal{X}(c), \Pi_\mathcal{X}(c) - c\rangle
\end{aligned}$$

which combined with the fact that $\|c - \Pi_\mathcal{X}(c)\|_2^2 \geq 0$ gives

$$\|x - \Pi_\mathcal{X}(c)\|_2^2 \leq \epsilon^2 - 2 \cdot \langle x - \Pi_\mathcal{X}(c), \Pi_\mathcal{X}(c) - c\rangle. \tag{A.2}$$

We will simply show that the inner product $\langle x - \Pi_\mathcal{X}(c), \Pi_\mathcal{X}(c) - c\rangle$ cannot be strictly negative to prove that $\|x - \Pi_\mathcal{X}(c)\|_2 \leq \epsilon$. Assume by contradiction that $\langle x - \Pi_\mathcal{X}(c), \Pi_\mathcal{X}(c) - c\rangle < 0$. Since $\Pi_\mathcal{X}(c) \in \mathcal{X}$ and $x \in \mathcal{X}$, it follows from the convexity of $\mathcal{X}$ that $\forall \lambda \in [0,1]$, $x_\lambda = \Pi_\mathcal{X}(c) + \lambda \cdot (x - \Pi_\mathcal{X}(c)) \in \mathcal{X}$. However, for all $\lambda \in (0,1)$ we have that

$$\begin{aligned}
\|x_\lambda - c\|_2^2 &= \|\Pi_\mathcal{X}(c) - c + \lambda \cdot (x - \Pi_\mathcal{X}(c))\|_2^2 \\
&= \|\Pi_\mathcal{X}(c) - c\|_2^2 + \lambda^2 \|x - \Pi_\mathcal{X}(c)\|_2^2 + 2\lambda \cdot \langle \Pi_\mathcal{X}(c) - c, x - \Pi_\mathcal{X}(c)\rangle \\
&= \|\Pi_\mathcal{X}(c) - c\|_2^2 + \lambda \cdot (\lambda \|x - \Pi_\mathcal{X}(c)\|_2^2 + 2 \cdot \langle \Pi_\mathcal{X}(c) - c, x - \Pi_\mathcal{X}(c)\rangle).
\end{aligned}$$

Therefore, taking any $0 < \lambda^\star < |\langle \Pi_\mathcal{X}(c) - c, x - \Pi_\mathcal{X}(c)\rangle| / \|\Pi_\mathcal{X}(c) - c\|_2^2 \wedge 1$ so that the second term of the right hand term of the previous equation is strictly negative gives that $\|x_{\lambda^\star} - c\|_2^2 < \|\Pi_\mathcal{X}(c) - c\|_2^2$ and leads us to the following contradiction $\min_{x \in \mathcal{X}} \|x - c\|_2 \leq \|x_{\lambda^\star} - c\|_2 < \|\Pi_\mathcal{X}(c) - c\|_2 = \min_{x \in \mathcal{X}} \|x - c\|_2$. Hence, $\langle x - \Pi_\mathcal{X}(c), \Pi_\mathcal{X}(c) - c\rangle \geq 0$ and we deduce from (A.2) that $\mathcal{X} \cap B(c,\epsilon) \subseteq B(\Pi_\mathcal{X}(c),\epsilon)$, which completes the proof. □

The next inequality will be useful to bound to bound volume of the intersection of a ball and a convex set.

Lemma A.4. *(From Zabinsky and Smith (1992), see Appendix Section therein). For any compact and convex set $\mathcal{X} \subset \mathbb{R}^d$ with non-empty interior, we have for any $x^\star \in \mathcal{X}$ and $\epsilon \in (0, \text{diam}(\mathcal{X}))$ that*

$$\frac{\mu(B(x^\star, \epsilon) \cap \mathcal{X})}{\mu(\mathcal{X})} \geq \left(\frac{\epsilon}{\text{diam}(\mathcal{X})} \right)^d.$$

Proof. We point out that a detailed proof of this result can be found in the Appendix Section of (Zabinsky and Smith (1992)). Nonetheless, we provide here a proof with less details for completeness. Introduce the similarity transformation $S : \mathbb{R}^d \to \mathbb{R}^d$ defined by

$$S : x \mapsto x^\star + \frac{r}{\text{diam}(\mathcal{X})}(x - x^\star)$$

and let $S(\mathcal{X}) := \{S(x) : x \in \mathcal{X}\}$ be the image of $\mathcal{X}$ by S. Since $x^\star \in \mathcal{X}$ and $\max_{x \in \mathcal{X}} \|x - x^\star\|_2 \leq \text{diam}(\mathcal{X})$ by definition, it follows from the convexity of $\mathcal{X}$ that $S(\mathcal{X}) \subseteq B(x^\star, r) \cap \mathcal{X}$ which implies that $\mu(B(x^\star, r) \cap \mathcal{X}) \geq \mu(S(\mathcal{X}))$. However, as S is a similarity transformation conserves the ratios of the volumes before/after transformation, we thus deduce that

$$\frac{\mu(B(x^\star, r) \cap \mathcal{X})}{\mu(\mathcal{X})} \geq \frac{\mu(S(\mathcal{X}))}{\mu(\mathcal{X})} = \frac{\mu(S(B(x^\star, \text{diam}(\mathcal{X}))))}{\mu(B(x^\star, \text{diam}(\mathcal{X})))} = \frac{\mu(B(x^\star, r))}{\mu(B(x^\star, \text{diam}(\mathcal{X})))}$$

and the result follows using the fact that $\forall r \geq 0$, $\mu(B(x^\star, r)) = \pi^{d/2} r^d / \Gamma(d/2 + 1)$ where $\Gamma(\cdot)$ stands for the standard gamma function. □

Lemma A.5. *Let $\mathcal{X} \subset \mathbb{R}^d$ be any convex set. Then, for any $\epsilon > 0$, the ϵ-ball of $\mathcal{X}$, defined by $B(\mathcal{X}, \epsilon) = \{x \in \mathbb{R}^d : \min_{x' \in \mathcal{X}} \|x - x'\|_2 \leq \epsilon\}$, is also a convex set.*

Proof. Pick any $(b_1, b_2) \in B(\mathcal{X}, \epsilon)^2$. By definition of $B(\mathcal{X}, \epsilon)$, we know that there exists $(x_1, \epsilon_1) \in \mathcal{X} \times B(\vec{0}, \epsilon)$ and $(x_2, \epsilon_2) \in \mathcal{X} \times B(\vec{0}, \epsilon)$ such that $b_1 = x_1 + \epsilon_1$ and $b_2 = x_2 + \epsilon_2$. Therefore, by convexity of $\mathcal{X}$ and $B(\vec{0}, \epsilon)$, we have that $\forall \lambda \in [0, 1]$,

$$(1 - \lambda) b_1 + \lambda b_2 \;=\; \underbrace{\lambda x_1 + (1 - \lambda) x_2}_{\in \mathcal{X}} \;+\; \underbrace{\lambda \epsilon_1 + (1 - \lambda) \epsilon_2}_{\in B(\vec{0}, \epsilon)}$$

which proves that $B(\mathcal{X}, \epsilon)$ is a convex set. □

B Analysis of the pure random search

In this section, we provide some results about the convergence of Pure Random Search that are used several times in the document. We start with the covering rate.

Proposition B.6. (Covering rate). *Let $X_1, \ldots, X_n$ be any sequence of n independent copies of $X \sim \mathcal{U}(\mathcal{X})$. Then, for any $\delta \in (0, 1)$, we have with probability at least $1 - \delta$,*

$$\sup_{x \in \mathcal{X}} \min_{i=1 \ldots n} \|X_i - x\|_2 \leq \text{diam}(\mathcal{X}) \cdot \left(\frac{\ln(n/\delta) + d \ln(d)}{n} \right)^{\frac{1}{d}}.$$

Proof. As the result trivially holds whenever $(\ln(n/\delta) + d\ln(d))/n \geq 1$, we consider the case where $(\ln(n/\delta) + d\ln(d))/n < 1$. From Proposition A.3, we know that there exists a sequence $x_1, \ldots, x_{N_\epsilon}$ of $N_\epsilon = \mathcal{N}_\epsilon(\mathcal{X})$ points in $\mathcal{X}$ such that $\mathcal{X} \subseteq \bigcup_{j=1}^{N_\epsilon} B(x_j, \epsilon)$. Hence, by using the bound on the covering number $\mathcal{N}_\epsilon(\mathcal{X})$ of Corollary A.3, we obtain that

$$
\begin{aligned}
\mathbb{P}\left(\sup_{x\in\mathcal{X}} \min_{i=1\ldots n} \|x - X_i\|_2 \leq \epsilon\right) &\geq \mathbb{P}\left(\bigcap_{j=1}^{N_\epsilon} \bigcup_{i=1}^{n} \{X_i \in B(x_j,\epsilon) \cap \mathcal{X}\}\right) \\
&= 1 - \mathbb{P}\left(\bigcup_{j=1}^{N_\epsilon} \bigcap_{i=1}^{n} \{X_i \notin B(x_j,\epsilon) \cap \mathcal{X}\}\right) \\
&\geq 1 - \sum_{j=1}^{N_\epsilon} \mathbb{P}\left(\bigcap_{i=1}^{n} \{X_i \notin B(x_j,\epsilon) \cap \mathcal{X}\}\right) \\
&\geq 1 - N_\epsilon \times \max_{j=1\ldots N_\epsilon} \mathbb{P}(X_1 \notin B(x_j,\epsilon) \cap \mathcal{X})^n \\
&= 1 - N_\epsilon \times \max_{j=1\ldots N_\epsilon} \left(1 - \frac{\mu(\mathcal{X} \cap B(x_j,\epsilon))}{\mu(\mathcal{X})}\right)^n \\
&\geq 1 - N_\epsilon \times \left(1 - \left(\frac{\epsilon}{\operatorname{diam}(\mathcal{X})}\right)^d\right)^n \\
&\geq 1 - \left(\frac{\sqrt{d}\operatorname{diam}(\mathcal{X})}{\epsilon}\right)^d \times \left(1 - \left(\frac{\epsilon}{\operatorname{diam}(\mathcal{X})}\right)^d\right)^n \\
&\geq 1 - \delta.
\end{aligned}
$$

□

The next result states the consistency property.

Proposition B.7. (Consistency equivalence). *Let $X_1, \ldots, X_n$ be a sequence of n independent random variables uniformly distributed over a compact and convex set $\mathcal{X} \subset \mathbb{R}^d$ and let $f : \mathcal{X} \to \mathbb{R}$ be any function admitting a global maximum over $\mathcal{X}$. Then, we have that*

$$
\max_{i=1\ldots n} f(X_i) \xrightarrow{p} \max_{x\in\mathcal{X}} f(x)
$$

if and only if the function f satisfies:

$$
\forall \epsilon > 0,\ \mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}) > 0.
$$

Proof. Noticing that $\forall n \in \mathbb{N}^\star$ and $\epsilon > 0$,

$$
\mathbb{P}\left(\max_{x\in\mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) > \epsilon\right) = \left(1 - \frac{\mu(\{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\})}{\mu(\mathcal{X})}\right)^n
$$

gives the result. □

The next result provides a bound on the covering rate.

Proposition B.8. (Convergence rates) *Let $\mathcal{X} \subset \mathbb{R}^d$ be any compact and convex set with non-empty interior and let $f : \mathcal{X} \to \mathbb{R}$ be any real-valued function admitting a unique optimizer $x^\star \in \arg\max_{x\in\mathcal{X}} f(x)$ and such that there exists $c_1, c_2 \geq 0$ and $\kappa_1 \geq \kappa_2 \geq 0$ for which we have $\forall x \in \mathcal{X}$:*

$$c_1 \cdot \|x - x^\star\|_2^{\kappa_1} \leq f(x^\star) - f(x) \leq c_2 \cdot \|x - x^\star\|_2^{\kappa_2}$$

Then, for any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$,

$$c_1 \cdot \mathrm{rad}(\mathcal{X})^{\kappa_1} \cdot \left(\frac{\delta}{2n}\right)^{\frac{\kappa_1}{d}} \leq \max_{x\in\mathcal{X}} f(x) - \max_{i=1\ldots n} f(X_i) \leq c_2 \cdot \mathrm{diam}(\mathcal{X})^{\kappa_2} \cdot \left(\frac{\ln(2/\delta)}{n}\right)^{\frac{\kappa_2}{d}}$$

where $X_1, \ldots, X_n$ is a sequence of n independent copies of $X \sim \mathcal{U}(\mathcal{X})$.

Proof. (Upper bound) Fix any $\delta \in (0,1)$, set $\epsilon = c_2 \mathrm{diam}(\mathcal{X})^{\kappa_2} (\ln(2/\delta)/n)^{\kappa_2/d}$ and let $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$ be the corresponding level set. Since the result trivially holds when $n \leq \ln(2/\delta)$, we consider the case where $n > \ln(2/\delta)$. Observe now that for all $x \in B(x^\star, (\epsilon/c_2)^{1/\kappa_2})) \cap \mathcal{X}$, we have that $|f(x) - f(x^\star)| \leq c_2 \cdot \|x - x^\star\|_2^{\kappa_2} = \epsilon$ which necessarily implies that $\mathcal{X} \cap B(x^\star, (\epsilon/c_2)^{1/\kappa_2}) \subseteq \mathcal{X}_\epsilon$. Therefore, it follows that

$$\begin{aligned}
\mathbb{P}\left(\max_{i=1\ldots n} f(X_i) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\right) &= \mathbb{P}\left(\bigcup_{i=1}^n \{X_i \in \mathcal{X}_\epsilon\}\right) && \text{(i)}\\
&= 1 - \mathbb{P}\left(X_1 \notin \mathcal{X}_\epsilon\right)^n && \text{(ii)}\\
&\geq 1 - \mathbb{P}(X_1 \notin \mathcal{X} \cap B(x^\star, (\epsilon/c_2)^{1/\kappa_2}))^n && \text{(iii)}\\
&= 1 - \left(1 - \left(\frac{\mu(\mathcal{X} \cap B(x^\star, (\epsilon/c_2)^{1/\kappa_2}))}{\mu(\mathcal{X})}\right)^d\right)^n && \text{(iv)}\\
&\geq 1 - \left(1 - \left(\frac{(\epsilon/c_2)^{1/\kappa_2}}{\mathrm{diam}(\mathcal{X})}\right)^d\right)^n && \text{(v)}\\
&= 1 - \left(1 - \frac{\ln(2/\delta)}{n}\right)^n && \text{(vi)}\\
&\geq 1 - \delta/2 && \text{(vii)}
\end{aligned}$$

where we used (i) the definition of $\mathcal{X}_\epsilon$, (ii) the fact that the random variable are independent, (iii) the fact that $\mathcal{X} \cap B(x^\star, \epsilon/k) \subseteq \mathcal{X}_\epsilon$, (iv) X_1 is uniformly distributed over $\mathcal{X}$, (v) Lemma A.4, (vi) the definition of ϵ and (vii) the elementary inequality $1 + x \leq e^x$.

(Lower Bound) Fix any $\delta \in (0,1)$, set $\epsilon = c_1 \mathrm{rad}(\mathcal{X})^{\kappa_1} (\delta/2n)^{\kappa_1/d}$ and let $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : f(x) \geq \max_{x\in\mathcal{X}} f(x) - \epsilon\}$ be the corresponding level set. Observe now that as for all $x \notin B(x^\star, \epsilon/c_1)^{1/\kappa_1}$, $f(x) \leq f(x^\star) - c_1 \|x - x^\star\|_2^{\kappa_1} < f(x^\star) - \epsilon$, we have that $\mathcal{X}_\epsilon \subseteq$

$B(x^\star, (\epsilon/c_1)^{1/\kappa_1})$. Hence, we have that

$$\begin{aligned}
\mathbb{P}\left(\max_{x\in\mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i) \leq \epsilon\right) &= \mathbb{P}\left(\bigcup_{i=1}^{n}\{X_i \in \mathcal{X}_\epsilon\}\right) && \text{(i)}\\
&\leq \mathbb{P}\left(\bigcup_{i=1}^{n}\left\{X_i \in B(x^\star, (\epsilon/c_1)^{1/\kappa_1})\right\}\right) && \text{(ii)}\\
&= 1 - \left(1 - \left(\frac{\mu(\mathcal{X}\cap B(x^\star, (\epsilon/c_1)^{1/\kappa_1}))}{\mu(\mathcal{X})}\right)\right)^n && \text{(iii)}\\
&\leq 1 - \left(1 - \left(\frac{(\epsilon/c_1)^{1/\kappa_1}}{\mathrm{rad}(\mathcal{X})}\right)^d\right)^n && \text{(iv)}\\
&= 1 - \left(1 - \frac{\delta}{2n}\right)^n && \text{(v)}\\
&\leq \delta/2 && \text{(vi)}
\end{aligned}$$

where we used (i) the definition of $\mathcal{X}_\epsilon$, (ii) the fact that $\mathcal{X}_\epsilon \subseteq B(x^\star, (\epsilon/c_1)^{1/\kappa_1})$, (iii) X_1 is uniformly distributed over $\mathcal{X}$, (iv) the volume of a ball, (v) the definition of ϵ and (vi) the last inequality provided at the beginning of the section. □

Proposition B.9. (Stopping time). *Let $\mathcal{X} \subset \mathbb{R}^d$ be any compact and convex set and let $f : \mathcal{X} \to \mathbb{R}$ be any continuous function with no flat parts (i.e. s.t. $\forall y \in \mathrm{Im}(f)$, $\mu(\{x \in \mathcal{X} : f(x) = y\}) = 0$). Then, if $\{X_i\}_{i\in\mathbb{N}^\star}$ denotes a sequence of independent random variables uniformly distributed over $\mathcal{X}$ and*

$$\tau_n = \inf\left\{m \in \mathbb{N}^\star : \max_{i=1\dots n+m} f(X_i) > \max_{i=1\dots n} f(X_i)\right\}$$

denotes the number of evaluations required to get an improvement after n iterations, for any $\delta \in (0, 1/2)$, we have with probability at least $1 - 2\delta$,

$$\left\lfloor \frac{\delta}{1-\delta}\cdot n\right\rfloor \leq \tau_n \leq \left\lceil \frac{1-\delta}{\delta}\cdot n\right\rceil$$

Proof. Observe that since $f(X_1), \dots, f(X_n)$ are n i.i.d. random variables and f has no flat parts, then $\hat{\imath}_n \sim \mathcal{U}([1, \dots, n])$ where $\hat{\imath}_n = \arg\max_{i=1\dots n} f(X_i)$. Therefore,

$$\begin{aligned}
\mathbb{P}(\tau_n > m) &= \mathbb{P}\left(\max_{i=n+1\dots n+m} f(X_i) < \max_{i=1\dots n} f(X_i)\right)\\
&= \mathbb{P}(\hat{\imath}_{n+m} \in \{1, \dots, n\})\\
&= \frac{n}{n+m}
\end{aligned}$$

and taking $m = \lfloor (1-\delta)n/\delta \rfloor$ and $m = \lfloor \delta n/(1-\delta) \rfloor$ gives the result. □

Proposition B.10. (Number of improvements). *Let $\{X_i\}_{i\in\mathbb{N}^\star}$ be a sequence of independent random variables uniformly distributed over $\mathcal{X}$ and let $f : \mathcal{X} \to \mathbb{R}$ be any continuous function with no flat parts (i.e. s.t. $\forall y \in \mathrm{Im}(f)$, $\mu(\{x \in \mathcal{X} : f(x) = y\}) = 0$). Then, if*

$$N_n = \sum_{i=1}^{n} \mathbb{I}\left\{f(X_{i+1}) > \max_{i=1\dots i} f(X_i)\right\}$$

denotes the number of improvements after n iterations, we have that $\forall n \in \mathbb{N}^\star$*:*

$$\ln(n+1) - 1 \leq \mathbb{E}[N_n] \leq \ln(n+1).$$

Proof. Using the same arguments as in the proof of Proposition B.9, we directly obtain that

$$\mathbb{E}[N_n] = \sum_{i=1}^{n} \mathbb{P}\left(\hat{\imath}_{i+1} = i+1\right) = \sum_{i=1}^{n} \frac{1}{i+1} = \sum_{i=1}^{n+1} \frac{1}{i} - 1$$

and the result follows using standard bounds on the harmonic series. □

C Optimization for gaussian processes via chaining

This section contains a preliminary work that has been presented in the NIPS Workshop on Bayesian Optimization (Contal et al. (2015)). It considers the stochastic optimization problem under a bandit feedback model, which is slightly different from the problem considered so far and generalizes the GP-UCB algorithm (Srinivas et al. (2010)) to arbitrary kernel and search space. We introduce a notion of covering trees and provide a novel optimization scheme based on the computation of covering numbers. The theoretical bounds we derive on the cumulative regret are more generic and present the same convergence rates as the GP-UCB algorithm. Finally, the algorithm is shown to be empirically as efficient as its natural competitors on real and synthetic tasks.

C. 1 Introduction

Optimizing an unknown and noisy function is at the center of many applications in the field of machine learning (Snoek et al. (2012)). We aim at finding the input of a system that returns the best output (or reward) by sequentially observing noisy evaluations of the function. The goal of an optimization procedure may be either seen as maximizing the sum of the rewards received at each iteration, that is to minimize the cumulative regret widely used in bandit problems, or as maximizing the best reward received so far, that is to minimize the simple regret. The task of designing efficient optimization procedures becomes challenging when we have very few information about the unknown function. To tackle this challenge, a Bayesian approach has been shown to be empirically efficient (Srinivas et al. (2012); Jones et al. (1998); Mockus (1989); Grünewälder et al. (2010); Roberts (2010); Krause and Ong (2011); Freitas et al. (2012); Hennig and Schuler (2012); Contal et al. (2013, 2014)). In this approach, to enforce near-by locations to have close associated values, the unknown function is modeled as a realization of a Gaussian Process (GP), which allows to control the assumptions we put on the smoothness of the function by choosing different kernels (Rasmussen and Williams (2006)). Our work is motivated by the fact that naive upper confidence bound (UCB) algorithms depend on the cardinality of a discretization of the search space. As a consequence, the convergence rates derived on the regret become arbitrary large when the discretization becomes finer and finer. In order to fill this gap between theory and practice, we propose an efficient computation of chaining for Bayesian Optimization. Chaining, that has been recently studied in the context of sequential optimization (see Grünewälder et al. (2010) for bandit with known horizon or Gaillard and Gerchinovitz (2015) in the case of online regression), appears to be an ideal

tool to capture both the impact of the smoothness of the function and the size of the search space. The contribution of this work is twofold: we first suggest a novel policy based on the computation of covering numbers called the CHAINING-UCB that can be seen as a generalization of the GP-UCB algorithm for arbitrary kernels and search space. On the other hand, we draw a theoretical analysis that leads us to upper bounds on its regret that have the same convergence rates as its competitors without depending on the cardinality of the discretization of the search space. The rest of the paper is organized as follows: in Section C. 2, we present the framework of our analysis, the basic properties of GP and we introduce the CHAINING-UCB algorithm. In Section C. 3, we present and discuss the main result of this work which is an upper bound on the regret. Finally, in Section C. 4 we compare the performances of the CHAINING-UCB algorithm to its natural competitors on real and synthetic problems. All the proofs are postponed to Section C. 6.

C. 2 The Chaining-UCB algorithm

Bayesian optimization framework. Let $f : \mathcal{X} \to \mathbb{R}$ be the unknown function modelling our underlying system, where the input space $\mathcal{X}$ is included in a convex subset of $\mathbb{R}^d$. In the sequel, we assume that f is a realization of a centered Gaussian process with known kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ and we use the notation $f \sim \mathcal{GP}(0,k)$. To avoid measurability issues we suppose that $\mathcal{X}$ is a finite set with arbitrary cardinality (see Boucheron et al. (2013); Talagrand (2014) for more details). We further suppose that $k(x,x') \leq 1$ for all $(x,x') \in \mathcal{X}^2$. A sequential optimization algorithm iterates two steps: it first chooses $X_1 \in \mathcal{X}$ and recieves $Y_1 = f(X_1) + \epsilon_1$ and then chooses X_t based on $Y_1, \dots, Y_{t-1}$ for $t \geq 2$, and next gets the noisy observation $Y_t = f(X_t) + \epsilon_t$ where $(\epsilon_t)_{t\geq 1}$ are independent Gaussians $\mathcal{N}(0,\eta^2)$ with known variance η^2. We denote by $\mathbf{X}_n = \{X_1, \dots, X_n\}$ the set of evaluation points after n iterations and by $\mathbf{Y}_n = [Y_1, \dots, Y_n]$ the associated observations packed in vector form. We denote by R_n the respective cumulative regret and by S_n the simple regret:

$$R_n = \sup_{x^\star \in \mathcal{X}} \sum_{t=1}^{n} f(x^\star) - f(X_t) \quad \text{and} \quad S_n = \sup_{x^\star \in \mathcal{X}} f(x^\star) - \max_{t=1\dots n} f(X_t),$$

on which we want to prove high probabilistic upper bounds. Compared to the work of Grünewälder et al. (2010) this paper considers that the time horizon n is unknown. For GP, the distribution of f conditioned on $\mathbf{Y}_n$ is a non-centered GP of mean μ_{n+1} and kernel k_{n+1} computed as follows for all $(x,x') \in \mathcal{X}^2$:

$$\mu_{n+1}(x) = \mathbf{k}_n(x)^\top \mathbf{C}_n^{-1} \mathbf{Y}_n \quad \text{and} \quad k_{n+1}(x,x') = k(x,x') - \mathbf{k}_n(x)^\top \mathbf{C}_n^{-1} \mathbf{k}_n(x'), \tag{C.3}$$

where $\mathbf{k}_n(x) = [k(x_t,x)]_{x_t \in \mathbf{X}_n}$ is the kernel vector between x and $\mathbf{X}_n$, and $\mathbf{C}_n = \mathbf{K}_n + \eta^2 \mathbf{I}$ with $\mathbf{K}_n = [k(x_t,x_{t'})]_{(x_t,x_{t'}) \in \mathbf{X}_n}$ the kernel matrix Rasmussen and Williams (2006). We also define the variance $\sigma_n^2(x) = k_n(x,x)$ and the pseudo-distance $d_n(x,x^\star)$:

$$d_n(x,x') = \sqrt{\sigma_n^2(x) + \sigma_n^2(x') - 2k_n(x,x')}. \tag{C.4}$$

Note that $d_n^2(x,x') = \mathrm{Var}[f(x) - f(x') \mid \mathbf{X}_n, \mathbf{Y}_n]$. To measure the "size" of $\mathcal{X}$ we will compute covering numbers with respect to d_n.

An upper confidence bound algorithm via chaining. At the core of our strategy to

Input: $n \in \mathbb{N}^\star$, $\mathcal{X}$, $\eta \geq 0$, $f : \mathcal{X} \to \mathbb{R}$

1. Initialization: Let $X_1 \sim \mathcal{U}(\mathcal{X})$
Evaluate $Y_1 = f(X_1) + \epsilon_1$, $t \leftarrow 1$

2. Iterations: Repeat while $t < n$:
Compute μ_t, σ_t, d_t
$T_0 \leftarrow \emptyset$, $\sigma_t^{\min} = \min_{x \in \mathcal{X}} \sigma_t(x)$
For $i = 1 \dots \lceil 1 - \log() \rceil$
$\epsilon_i \leftarrow 2^{-i+1}$
$\bar{\mathcal{X}} \leftarrow \{x \in \mathcal{X} : d_t(x, T_{i-1}) > \epsilon_i\}$
$T_i \leftarrow T_{i-1} \cup \text{Cover}(\bar{\mathcal{X}}, d_t, \epsilon_i)$
$H_i \leftarrow \epsilon_i \sqrt{2 \log\left((|T_i| + 1) i^2 t^2 \pi^4 / (36\delta)\right)}$
$X_t \leftarrow \arg\max_{x \in \mathcal{X}} \ \mu_t(x) + \sum_{i : \sigma_t^{\min} \leq \epsilon_i \leq \sigma_t(x)}$
Evaluate $Y_t = f(X_t) + \epsilon_t$, $t \leftarrow t + 1$

3. Output: Return $X_{\hat{\imath}_n}$ where $\hat{\imath}_n \in \arg\max_{i=1 \dots n} Y_i$

Figure C.1: The Chaining-UCB algorithm

Input: $\mathcal{X}$, $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, $\epsilon \geq 0$

1. Initialization: $T \leftarrow \emptyset$, $\bar{\mathcal{X}} \leftarrow \mathcal{X}$
$\forall (x, x') \in \mathcal{X}^2$, $G[x, x'] \leftarrow \mathbb{I}\{d(x, x') \leq \epsilon\}$

2. Iterations: While $\bar{\mathcal{X}} \neq \emptyset$:
$x \leftarrow \arg\max_{x \in \bar{\mathcal{X}}} \sum_{x' \in \bar{\mathcal{X}}} G[x, x']$
$T_0 \leftarrow \emptyset$, $\sigma_t^{\min} = \min_{x \in \mathcal{X}} \sigma_t(x)$
$T \leftarrow T \cup \{x\}$
$\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \setminus \{x' \in \bar{\mathcal{X}} : G[x, x'] = 1\}$

3. Output: Return T

Figure C.2: The Greedy-cover algorithm

control the regret of the algorithm we need to prove an upper confidence bound (UCB) on $\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x)$ for all $x \in \mathcal{X}$. A naive approach uses a union bound on $\mathcal{X}$, resulting in a factor $\sqrt{\log |\mathcal{X}|}$ in the UCB, which is not appropriate when $\mathcal{X}$ is a numerical discretization of a continuous space, typically a multi-dimensional grid of high density. We use the chaining trick (Talagrand (2014); Pollard (1990); Dudley (1967)) to get a UCB relying on the covering numbers of $\mathcal{X}$ with respect to d_n instead of its cardinality. In that way the UCB does not increase when we refine the discretization of any continuous space. The main element of our algorithm is the computation of an ϵ-cover of $\mathcal{X}$. Here, we say that a subset T is an ϵ-cover of $\mathcal{X}$ when for all $x \in \mathcal{X}$, $d_n(x, T) \leq \epsilon$, where $d_n(x, T) = \inf_{x' \in T} d_n(x, x')$. The covering numbers $N(\mathcal{X}, d_n, \epsilon)$ are the cardinality of the smallest ϵ-cover of $\mathcal{X}$ for the pseudo-distance d_n, and the function $\textsc{Cover}(\mathcal{X}, d_n, \epsilon)$ of the

Greedy algorithm of Figure C.2 returns such a set. The CHAINING-UCB algorithm then queries the objective function at the point maximizing the UCB obtained by chaining. The computation of an optimal ϵ-cover is NP-hard, but we can easily build an efficient approximation as discussed in Section C. 4.

C. 3 Theoretical guarantees

Guarantees on the regret. In the following theorem we describe a high probabilistic upper bound on the instantaneous regret incurred by the Chaining-UCB algorithm (Figure C.1). This inequality is used in the subsequent corollary to obtain upper bounds on its cumulative and simple regrets. The proof of Theorem C.11 requires to define several theoretical quantities, it is postponed to Section C. 6.

Theorem C.11. (GENERIC UPPER BOUND) *Let $\mathcal{X}$ be any finite space with arbitrary large cardinality included in a convex subset of $\mathbb{R}^d$ and let $X_1, X_2, \dots$ be a series of evaluation points generated by the CHAINING-UCB algorithm on f sampled from a GP$(0,k)$ where $k(\cdot,\cdot) \leq 1$. Then, for any $\delta \in (0,1)$, using the notations $\sigma_n = \sigma_n(x_n)$ and $a_{n,\delta} = 6\sqrt{\log \frac{n^2\pi^4}{36\delta}} + 15$, there exist constants $b < 6$ and $c < 9$ such that with probability at least $1-\delta$, we have for all $n \geq 1$:*

$$\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x_n) \leq \sigma_n\big(a_{n,\delta} - b\log\sigma_n\big) + c \sum_{i:2^{1-i}<\sigma_n} 2^{1-i}\sqrt{\log N(\mathcal{X}, d_n, 2^{1-i})}\,.$$

Here, the covering numbers $\mathcal{N}(\mathcal{X}, d_n, \epsilon_i)$ measure the size of the search space $\mathcal{X}$ with respect to the pseudo-distance d_n. In order to simplify this inequality and get convergence rates for the cumulative and simple regrets, it is necessary to add some assumptions on d_n and $\mathcal{X}$. Corollary C.12 gives an example of the rates we obtain for the usual Squared-Exponential kernel $k(x,x') = \exp(-\|x-x'\|_2^2/2)$, where the pseudo-distance enjoys a Lipschitz property and the covering numbers can be further bounded. Indeed, we have in this respect that $d(x,x') \leq \|x-x'\|_2$ for all $(x,x') \in \mathcal{X}\times\mathcal{X}$. By definition of d_n we know that $d_n(x,x') \leq d(x,x')$ for all $n \geq 1$. When $\mathcal{X} \subseteq [0,R]^D$, it follows that $N(\mathcal{X}, d_n, \varepsilon) \leq (\frac{R}{\epsilon})^D$. We show that both the cumulative and simple regret possess upper bounds matching with the current state of the art, as specified in Corollary C.12.

Corollary C.12. *For the Squared-Exponential kernel and $\mathcal{X} \subseteq [0,R]^D$ the CHAINING-UCB algorithm returns a sequence $X_1, X_2, \dots, X_n$ such that for any $\delta \in (0,1)$, we have with probability at least $1-\delta$:*

$$R_n = O\Big(\sqrt{n(\log n)^{D+2}}\Big) \quad \text{and} \quad S_n = O\Big(\sqrt{\frac{(\log n)^{D+2}}{n}}\Big).$$

To prove Corollary C.12 we first use the characterization of the covering numbers to obtain that $R_n = O\Big(\sum_{t=1}^n \sigma_t\sqrt{\log t} - \sigma_t\log\sigma_t\Big)$. We then invoke the inequality proven in Theorem 5 of Srinivas et al. (2012) which shows that $\sum_{t=1}^n \sigma_t^2 = O\big((\log n)^{D+1}\big)$. We see by maximizing R_n under this constraint with Lagrange multipliers that $R_n = O\big(\sqrt{n(\log n)^{D+2}}\big)$. The rate for S_n directly follows from this result. It is straightforward to apply this technique to other cases like linear kernels or Matérn kernels with parameter $\nu > 2$, since both the covering numbers and $\sum_{t=1}^n \sigma_t^2$ can be bounded by similar

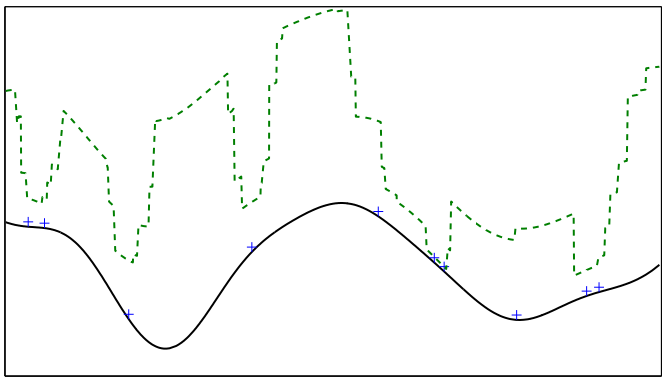


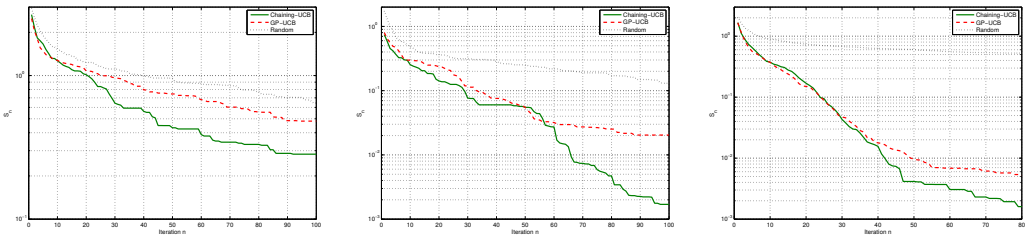
Figure C.3: Illustration of the UCB used in the Chaining-UCB algorithm (Figure C.1). The plain black line is the target function f. The blue crosses are the noisy observations. The dashed green line is the UCB used by the CHAINING-UCB algorithm. The rectangular form of the UCB is explained by the discrete sum. The next query selected by the algorithm is the maximum of the UCB.

results.

A practical algorithm A reader familiar with classical chaining may ask why we never use the Dudley integral $\int_0^{\sigma_n} \sqrt{\log N(\mathcal{X}, d_n, \epsilon)}\, d\epsilon$ instead of the discrete sum $\sum_{i:\epsilon_i<\sigma_n} \epsilon_i \sqrt{\log N(\mathcal{X}, d_n, \epsilon_i)}$. The main reason is purely practical. Even if the Dudley integral is simple to bound for certain kernel k and space $\mathcal{X}$, we want the Chaining-UCB algorithm (Figure C.1) to be able to adapt to all configurations without having to tune its parameters. By computing the covering numbers, the CHAINING-UCB algorithm calibrates automatically the exploration-exploitation tradeoff. This fact contrasts with previous algorithms like GP-UCB where the input parameter β_t depends either on the cardinality $|\mathcal{X}|$ or on the Lipschitz constants of the kernel (see Theorems 1 and 2 of Srinivas et al. (2012)). By using the discrete sum instead of the integral, we also limit the number of ϵ_i-covers which need to be computed to only the ϵ_i such that $\epsilon_i > \min_{x\in\mathcal{X}} \sigma_n(x)$. Thanks to their geometrical decay, these numbers remain low in practice. Figure C.3 illustrates the UCB computed with this discrete sum on a toy example in one dimension.

C. 4 Practical considerations and experiments

Computing the ϵ-covers using an algorithm on graph. As mentioned previously the computation of an optimal ϵ-cover is NP-hard. We demonstrate here how to build an efficient approximation in practice using a greedy algorithm on graph. We first remark that for a fixed ϵ we can define a graph G where the nodes are the elements of $\mathcal{X}$ and there is an edge between x and x' if and only if $d(x, x') \leq \epsilon$. The size of this construction is $O(|\mathcal{X}|^2)$. It is possible to exploit the sparse structure of the underlying graph to get an efficient representation. The problem of finding an optimal ϵ-cover reduces to the problem of finding a minimal dominating set on G. We can therefore use the greedy Algorithm (Figure C.2) which enjoys an approximation factor of $\log(d_{\max}(G))$, where $d_{\max}(G)$ is the maximum degree of G (see for example Johnson (1973) for a proof of NP-



(a) SE kernel (b) Himmelblau (c) Graph kernel

Figure C.4: Empirical mean of the simple regret S_n in terms of iteration n for CHAINING-UCB, GP-UCB and RANDOM search (lower is better) on (a) SE kernel, (b) Himmeblau and (c) Graph kernel .

hardness and approximation results). This leads to an additional (almost constant) term of $\sqrt{\log\log d_{\max}(G)}$ in the upper bound of Theorem C.11. Note that this approximation is optimal unless $P = NP$ as shown in Raz and Safra (1997).

Experiments. In this section we compare the ability of the CHAINING-UCB algorithm to find the maximum of an unknown function against the GP-UCB algorithm from Srinivas et al. (2012) and the RANDOM search. For both the CHAINING-UCB and the GP-UCB algorithms the value for δ was set to 0.05. The RANDOM approach selects the next query uniformly among the unknown locations. It gives a baseline to grasp the scale of the performances of both algorithms. All three strategies are initialized with a set of 10 noisy observations sampled uniformly over $\mathcal{X}$. Figure C.4 shows the empirical mean of the simple regret S_n over 32 runs. In every experiments the standard deviation of the noise was set to 0.05 and the search space is discrete with $|\mathcal{X}| = 10^4$. The SE experiment consists in generating GPs drawn from a two dimensional isotropic SE kernel, with the kernel bandwidth set to 1. The search space is a uniform design in a square of length 20. The Himmelblau experiment is a two dimensional polynomial function based on the Himmelblau's function with the addition of a linear trend. It possesses several local maxima which makes the global optimization challenging. The kernel used by the algorithm is an isotropic SE kernel with the bandwidth chosen by maximizing the marginal likelihood of a subset of training points. Finally we consider the task of optimizing over graphs. Global optimization of graphs can model complex problems as in industrial design, network analysis or computational biology. The input space is the set of directed graphs with less than 20 nodes. The kernel is the shortest-path kernel (Borgwardt and Kriegel (2005)) normalized such that $k(g,g) = 1$ for $g \in \mathcal{X}$. Note that in this synthetic assessment we do not address the question faced in practice of choosing the prior. We further mention that the kernel matrix can be efficiently computed by pre-processing all pairs of shortest paths for each graph with Floyd-Warshall's algorithm. Figure C.4 shows that the CHAINING-UCB algorithm is empirically more efficient than the GP-UCB algorithm on the three test cases. We remark that in practice, unlike GP-UCB, we may use a design with $|\mathcal{X}| \gg 10^4$ without affecting the performance of CHAINING-UCB. However generating a GP costs $O(|\mathcal{X}|^3)$ which limits the tractability of synthetic assessments. Finally, we remark that the CHAINING-UCB algorithm is empirically as efficient as the GP-UCB algorithm on the generated GP challenge, and significantly better when the underlying function is not sampled from a GP. This suggests that the automatic calibration of the exploration/exploitation tradeoff by

the covering tree effectively adapt to various settings.

C. 5 Conclusion

We have introduced a sequential optimization strategy based on the computation of covering numbers to automatically calibrate the exploration-exploitation tradeoff. Statistical guarantees are provided in terms of convergence rates for both the cumulative and simple regrets. We demonstrate empirically that the algorithm performs as well as its natural competitor on various test functions. To the best of our knowledge, this is the first time covering numbers are computed in an efficient sequential optimization algorithm. This contribution achieves a step toward theoretical lower bounds for GP optimization. We intend to extend this work to build an algorithm based on generic chaining (Talagrand (2014)). The majorizing measure theorem may then provide lower bound on the supremum of the regret.

C. 6 Proofs

In this section, we first demonstrate how chaining is used to analyze the supremum of a centered GP. Since our aim is to derive an efficient algorithm, we put some effort to keep low values for the constants. The next subsection shows how this simply extends to non-centered GP and how to build the UCB algorithm with the chaining bound. Finally we express the upper bounds in terms of the covering numbers of $\mathcal{X}$.

Using covering trees to bound the supremum of a GP. Let us fix an iteration n and the observations $\mathbf{Y}_{n-1}$, we define f_n the centered posterior of f conditioned on $\mathbf{Y}_{n-1}$, which is a Gaussian process $\mathcal{GP}(0, k_n)$ where k_n is defined in Equation C.3. By Cramer-Chernoff's method (Boucheron et al. (2013)) we have for all $(x^\star, x) \in \mathcal{X}^2$ and $u > 0$ that:

$$\mathbb{P}\left(f_n(x^\star) - f_n(x) > d_n(x^\star, x)\sqrt{2u}\right) < e^{-u}, \tag{C.5}$$

where the pseudo-distance d_n is defined Equation C.4. We use classical chaining (Talagrand (2014); Dudley (1967)) to obtain concentration inequalities on $\sup_{x^\star \in \mathcal{X}} f_n(x^\star) - f_n(x)$ with respect to the covering numbers. The idea of chaining performs as follows. Let $\pi_i : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ be a sequence of mappings such that $\pi_0(x, x^\star) = x$ and $\pi_i(x, x^\star) \to_{i\to\infty} x^\star$ holds for all $x, x^\star \in \mathcal{X}$. We then have the telescopic sum,

$$\sup_{x^\star \in \mathcal{X}} f_n(x^\star) - f_n(x) = \sup_{x^\star \in \mathcal{X}} \sum_{i \geq 1} f_n\big(\pi_i(x, x^\star)\big) - f_n\big(\pi_{i-1}(x, x^\star)\big)\,. \tag{C.6}$$

Since $|\mathcal{X}|$ may be arbitrary large we cannot perform a union bound over all $x^\star$ in Equation C.6. Instead we enforce π_i to take its values in a discrete subset T_i of $\mathcal{X}$ formed with carefully chosen points such that $d(\pi_i(x, x^\star), \pi_{i-1}(x, x^\star))$ is small, so we can use Equation C.5. By controlling $|T_i|$ we are able to achieve a union bound over T_i and i. We describe here the sequence $\{T_i\}_{i\geq 0}$ of subsets of $\mathcal{X}$, which satisfy $T_0 = \{x_0\}$, $T_i \subseteq T_{i+1}$ for all $i \geq 0$ and $\mathcal{X} = \bigcup_{i\geq 0} T_i$. For all level i we fix an exponentially decreasing radius $\epsilon_i = 2^{-i+1}$ and define T_i as the smallest ϵ_i-cover of $\mathcal{X}$ containing T_{i-1}:

$$T_i \in \underset{\substack{T \supseteq T_{i-1} \\ d_n(x,T) \leq \epsilon_i \ \forall x \in \mathcal{X}}}{\operatorname{argmin}} |T|\,. \tag{C.7}$$

The usual analysis of the chaining upper bound is not suitable for sequential optimization since our aim is to bound $\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x)$ for a specific x chosen by the algorithm. In this respect, we focus on finer results which keep track on the distance $d(x,x^\star)$. Unfortunately, this needs to define additional theoretical quantities and pay more attention to the sharpness of the inequalities involved. We endow $\mathcal{X}$ with a tree structure with respect to $\{T_i\}_{i\geq 0}$ and d_n where the root is x_0 and the parent of an element of T_{i+1} is its closest point among the elements of T_i. Formally the parent relationship $p : \mathcal{X} \to \mathcal{X}$ satisfies that for all $x \in T_{i+1} \setminus T_i$, $p(x) \in \operatorname{argmin}_{x'\in T_i} d(x,x')$ and $p(x_0) = x_0$ for the root. We denote by $p_i(x)$ the ancestor of x at level i, that is the element x_i of T_i such that there exists a branch $x_i, x_{i+1}, \ldots, x$ where $p(x_{j+1}) = x_j$ for $j > i$. The sequence $\{p_i(x^\star)\}_{i\geq 0}$ is illustrated on Figure C.5. The path from x to $x^\star$ given by $\pi_i(x,x^\star)$ works as follows: we start and stay on x until one parent of $x^\star$ at level i is not too far. Precisely for each $x, x^\star \in \mathcal{X}$ and level i, $\pi_i(x,x^\star)$ is either $p_i(x^\star)$ or x:

$$\pi_i(x,x^\star) = \begin{cases} p_i(x^\star) & \text{if } \epsilon_i < d_n(x,x^\star)\,, \\ x & \text{otherwise}\,. \end{cases}$$

Figure C.5 highlights the path defined by $\pi_i(x,x^\star)$. Using the definition of T_i and π_i we have $d_n\big(\pi_i(x,x^\star), \pi_{i-1}(x,x^\star)\big) \leq \epsilon_{i-1}$, and if $\epsilon_i \geq d_n(x,x^\star)$ then $\pi_i(x,x^\star) = x$ and the i-th summand in Equation C.6 is equal to zero. Now using the concentration inequality of Equation C.5 we get for all $u_i > 0$:

$$\mathbb{P}\Big(f_n\big(\pi_i(x,x^\star)\big) - f_n\big(\pi_{i-1}(x,x^\star)\big) > \epsilon_{i-1}\sqrt{2u_i}\Big) < e^{-u_i}\,.$$

Thanks to the tree structure $\forall x \in \mathcal{X}, i \geq 1$, $\Big|\Big\{\big(\pi_i(x,x^\star), \pi_{i-1}(x,x^\star)\big) : x^\star \in \mathcal{X}\Big\}\Big| = |T_i| + 1$. We obtain for all $u_i > 0$:

$$\mathbb{P}\left(\bigcup_{x^\star\in\mathcal{X}}\bigcup_{i>0}\Big\{f_n\big(\pi_i(x,x^\star)\big) - f_n\big(\pi_{i-1}(x,x^\star)\big) > \epsilon_{i-1}\sqrt{2u_i}\Big\}\right) < \sum_{i>0}(|T_i|+1)e^{-u_i}\,.$$

For any $u > 0$, we set $u_i = u + \log\big((|T_i|+1)i^2\frac{\pi^2}{6}\big)$, so that $\sum_{i>0}(|T_i|+1)e^{-u_i} = e^{-u}$. Combining the previous union bound with the chaining property we have a generic upper bound:

$$\forall x \in \mathcal{X}, \mathbb{P}\left(\sup_{x^\star\in\mathcal{X}} f_n(x^\star) - f_n(x) > \sup_{x^\star\in\mathcal{X}} \sum_{i:\epsilon_i < d(x,x^\star)} \epsilon_{i-1}\sqrt{2u_i}\right) < e^{-u}\,. \tag{C.8}$$

From classical chaining to optimization. The previous inequality provides us with a powerful tool to prove upper bounds on the regret. We use two steps to be able to build the algorithm. We first adapt this result to non-centered processes. We then split the sum in two terms, one for x and one for $x^\star$, and we demonstrate that the queries made by the Chaining-UCB algorithm of Figure C.1 allow to control both terms. At iteration n, the posterior process of f conditioned on $\mathbf{Y}_{n-1}$ has a mean μ_n and covariance k_n defined in Equation C.3. We fix $u_n > 0$ and we note $\{T_i^n\}_{i\geq 1}$ the covering tree with respect to d_n defined in the previous subsection. We denote $H_i^n = \epsilon_{i-1}\sqrt{2u_i^n}$ the associated summands of Equation C.8, where $u_i^n = u_n + \log\big((|T_i^n|+1)i^2\frac{\pi^2}{6}\big)$. In order to obtain an analogous

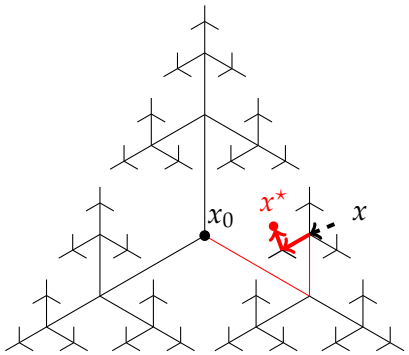


Figure C.5: Illustration of the tree structure. The root of the tree is x_0, the 3 neighbors of x_0 form T_1, the next 9 neighbors of T_1 form T_2 and so on. The red edges connect the branch $\{p_i(x^\star)\}_{i\leq 4}$ and the thick arrows show the points $\{\pi_i(x,x^\star)\}_{i\leq 4}$. Note that the positions of the nodes are arbitrary and do not form an ϵ-cover with respect to the euclidean distance.

result for f, it suffices to replace $f_n(\cdot)$ in Equation C.6 by the centered process $f(\cdot)-\mu_n(\cdot)$ and remark that the terms $\mu_n(\cdot)$ collide, we get:

$$\forall x\in\mathcal{X},\ \mathbb{P}\left(\sup_{x^\star\in\mathcal{X}} f(x^\star)-f(x)-\mu_n(x^\star)+\mu_n(x) > \sup_{x^\star\in\mathcal{X}}\sum_{i:\epsilon_i<d_n(x,x^\star)} H_i^n\right) < e^{-u_n}\,.$$

With some abuse of notation, we write in the sequel $\sigma_n=\sigma_n(x_n)$ and $\sigma_n^\star=\sigma_n(x^\star)$. We now decompose the indices of the sum in two sets and take the supremum on both:

$$\sup_{x^\star\in\mathcal{X}}\sum_{i:\epsilon_i<d_n(x,x^\star)} H_i^n \leq \sup_{x^\star\in\mathcal{X}}\sum_{i:\epsilon_i<\sigma_n^\star} H_i^n + \sup_{x^\star\in\mathcal{X}}\sum_{\substack{i:\sigma_n^\star\leq\epsilon_i\\ \epsilon_i<d_n(x,x^\star)}} H_i^n\,.$$

Since for the Chaining-UCB algorithm we have $x_n\in\operatorname{argmax}_{x\in\mathcal{X}}\mu_n(x)+\sum_{i:\epsilon_i<\sigma_n(x)}H_i^n$, we obtain for all $x^\star\in\mathcal{X}$ that $\mu_n(x^\star)-\mu_n(x_n)\leq\sum_{i:\epsilon_i<\sigma_n}H_i^n-\sum_{i:\epsilon_i<\sigma_n^\star}H_i^n$. We thus prove with probability at least $1-e^{-u_n}$ that:

$$\sup_{x^\star\in\mathcal{X}} f(x^\star)-f(x_n) \leq \sum_{i:\epsilon_i<\sigma_n} H_i^n + \sup_{x^\star\in\mathcal{X}}\sum_{\substack{i:\sigma_n^\star\leq\epsilon_i\\ \epsilon_i<d_n(x_n,x^\star)}} H_i^n\,. \tag{C.9}$$

We simplify this inequality by checking that the second sum on the right side is smaller than twice the first sum. This involves tedious but elementary calculus using the geometric decay of ϵ_i, the linearity of H_i^n in ϵ_{i-1} and the fact that $d_n(x_n,x^\star)\leq\sigma_n+\sigma_n^\star$. It suffices to verify that for any $(a,b)\in[0,1]^2$, $\sum_{i:a\leq 2^{-i}\leq a+b}H_i^n\leq 2\sum_{i:2^{-i}\leq b}H_i^n$. Without loss of generality we consider that $b\leq a$. Let $i_a=-\lceil\log_2 a\rceil$, $i_b=-\lfloor\log_2 b\rfloor$ and $i_{a,b}=-\lfloor\log_2(a+b)\rfloor$. Since $\epsilon_i=2(\epsilon_i-\epsilon_{i+1})$ and u_i^n increases as i increases, we have on one side $\sum_{i=i_b}^{\infty}\epsilon_i\sqrt{2u_i^n}\geq 2\frac{b}{2}\sqrt{2u_{i_b}^n}$, and on the other side $\sum_{i=i_{a,b}}^{i_a}\epsilon_i\sqrt{2u_i^n}\leq 2b\sqrt{2u_{i_a}^n}$. With $u_{i_a}^n\leq u_{i_b}^n$ we get $\sum_{i=i_{a,b}}^{i_a}\epsilon_i\sqrt{2u_i^n}\leq 2\sum_{i=i_b}^{\infty}\epsilon_i\sqrt{2u_i^n}$. Plugging this result in Equation C.9 we have with probability at least $1-e^{-u_n}$ that $\sup_{x^\star\in\mathcal{X}}f(x^\star)-f(x_n)\leq 3\sum_{i:\epsilon_i<\sigma_n}H_i^n$.

If we set u_n to $\log(n^2\frac{\pi^2}{6\delta})$ for any $\delta > 0$, that is $u_i^n = \log\Big((|T_i^n|+1)i^2n^2\frac{\pi^4}{\delta 6^2}\Big)$, we prove by a union bound over all $n \geq 1$ that with probability at least $1-\delta$:

$$\forall n > 0,\ \sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_n) \leq 3\sum_{i:\epsilon_i<\sigma_n} \epsilon_{i-1}\sqrt{2u_i^n}\,. \tag{C.10}$$

Bounding with covering numbers. In order to relate Equation C.10 to the size of the input space we bound the cardinality $\big|T_i^n\big|$ by the covering numbers of $\mathcal{X}$. In fact by construction of T_i^n we know that $\big|T_i^n\big| + 1 \leq N(\mathcal{X},d_n,\epsilon_i) + N(\mathcal{X},d_n,\epsilon_{i-1}) + 1$, which is less than $3N(\mathcal{X},d_n,\epsilon_i)$. Note that the factor 3 is far from being tight for large $|\mathcal{X}|$, but the CHAINING-UCB algorithm does not need this crude upper bound since it only uses $\big|T_i^n\big|$. Splitting u_i^n in three factors, the LHS of Equation C.10 is smaller than $3\sum_{i:\epsilon_i<\sigma_n}\epsilon_{i-1}\Big(\sqrt{2\log(\big|T_i^n\big|+1)} + 2\sqrt{\log i} + \sqrt{\log\frac{n^2\pi^4}{36\delta}}\Big)$. Using the fact that $6\sqrt{2\log 3} < 9$ and that for any $s \in [0,1]$, $\sum_{i:2^{-i}<s} 2^{-i}\sqrt{\log i} < s - s\log s$ and $\sum_{i:2^{-i}<s} 2^{-i} < 2s$, we finally get that with probability at least $1-\delta$:

$$\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_n) \leq 9\sum_{i:\epsilon_i<\sigma_n}\epsilon_i\sqrt{\log N(\mathcal{X},d_n,\epsilon_i)} + 6\sigma_n\sqrt{\log\frac{n^2\pi^4}{36\delta}} + 15\sigma_n - 6\sigma_n\log\sigma_n\,,$$

proving Theorem C.11.

Proof of Corollary C.12. To prove Corollary C.12 we first remark that for the SE kernel $k(x,x') = e^{-\frac{1}{2}\|x-x'\|_2^2}$ the pseudo-distance enjoys $d(x,x') \leq \|x-x'\|_2$ for all $x,x' \in \mathcal{X}$. By definition of d_n we know that $d_n(x,x') \leq d(x,x')$ for all $n \geq 1$. When $\mathcal{X} \subseteq [0,R]^D$, it follows that $N(\mathcal{X},d_n,\varepsilon) \leq (\frac{R}{\epsilon})^D$. Plugging this into Theorem C.11 we obtain that $R_n = O\Big(\sum_{t=1}^n \sigma_t\sqrt{\log t} - \sigma_t\log\sigma_t\Big)$. We then invoke the inequality proven in Theorem 5 of Srinivas et al. (2012) which shows that $\sum_{t=1}^n \sigma_t^2 = O\big((\log n)^{D+1}\big)$. We see by maximizing R_n under this constraint with Lagrange multipliers that $R_n = O\big(\sqrt{n(\log n)^{D+2}}\big)$. The rate for S_n directly follows from this result.

Bibliography

Anderssen, R. and P. Bloomfield

1975. Properties of the random search in global optimization. *Journal of Optimization Theory and Applications*, 16(5):383–398.

Audet, C. and J. E. Dennis Jr

2006. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217.

Auger, A.

2005. Convergence results for the (1, λ)-sa-es using the theory of ϕ-irreducible markov chains. *Theoretical Computer Science*, 334(1-3):35–69.

Avriel, M.

2003. *Nonlinear programming: analysis and methods*. Courier Corporation.

Barton, R. R.

1994. Metamodeling: a state of the art review. In *Simulation Conference Proceedings*, pp. 237–244. IEEE.

Bélisle, C. J., H. E. Romeijn, and R. L. Smith

1993. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2):255–266.

Bergstra, J. and Y. Bengio

2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bethke, A. D.

1978. *Genetic algorithms as function optimizers*. PhD thesis, University of Michigan, Ann Arbor.

Birgé, L. and P. Massart

1997. From model selection to adaptive estimation. In *Festschrift for lucien le cam*, pp. 55–87. Springer.

Bobkov, S.

2010. On concentration of measure on the cube. *Journal of Mathematical Sciences*, 165(1):60–70.

Boneh, A. and A. Golan

1979. Constraints redundancy and feasible region boundedness by random feasible point generator (rfpg). In *Third European congress on operations research, Amsterdam*.

Borgwardt, K. M. and H.-P. Kriegel

2005. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pp. 8–pp. IEEE.

Boucheron, S., G. Lugosi, and P. Massart
2013. *Concentration inequalities: A nonasymptotic theory of independence*. OUP Oxford.

Bousquet, O., S. Gelly, K. Kurach, O. Teytaud, and D. Vincent
2017. Critical hyper-parameters: No random, no cry. *arXiv preprint arXiv:1706.03200*.

Boyd, S. and L. Vandenberghe
2004. *Convex optimization*. Cambridge University Press.

Brent, R. P.
1973. *Algorithms for minimization without derivatives*. Prentice-Hall, Englewood Cliffs.

Brooks, S. H.
1958. A discussion of random methods for seeking maxima. *Operations research*, 6(2):244–251.

Bubeck, S., G. Stoltz, and J. Y. Yu
2011. Lipschitz bandits without the lipschitz constant. In *International Conference on Algorithmic Learning Theory*, pp. 144–158. Springer.

Bull, A. D.
2011. Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904.

Carr, J. M., D. Mazauric, F. Cazals, and D. J. Wales
2016. Energy landscapes and persistent minima. *The Journal of chemical physics*, 144(5):054109.

Cavalier, L.
1997. Nonparametric estimation of regression level sets. *Statistics A Journal of Theoretical and Applied Statistics*, 29(2):131–160.

Cerf, R.
1994. *Une théorie asymptotique des algorithmes génétiques*. PhD thesis, Université Montpellier II.

Clémençon, S.
2011. On U-processes and clustering performance. In *Advances in Neural Information Processing Systems*, pp. 37–45.

Clémençon, S., G. Lugosi, and N. Vayatis
2010. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, pp. 844–874.

Clémençon, S. and N. Vayatis
2010. Overlaying classifiers: a practical approach to optimal scoring. *Constructive Approximation*, 32(3):619–648.

Cohn, D., L. Atlas, and R. Ladner
1994. Improving generalization with active learning. *Machine learning*, 15(2):201–221.

Conn, A. R., K. Scheinberg, and P. L. Toint
1997. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical programming*, 79(1):397–414.

Conn, A. R., K. Scheinberg, and L. N. Vicente
2009. *Introduction to derivative-free optimization.* SIAM.

Contal, E., D. Buffoni, A. Robicquet, and N. Vayatis
2013. Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–240. Springer.

Contal, E., C. Malherbe, and N. Vayatis
2015. Optimization for gaussian processes via chaining. In *NIPS Workshop on Bayesian Optimization.*

Contal, E., V. Perchet, and N. Vayatis
2014. Gaussian process optimization with mutual information. In *International Conference on Machine Learning*, pp. 253–261.

Dasgupta, S.
2011. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781.

Dasgupta, S., C. Monteleoni, and D. J. Hsu
2007. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, pp. 353–360.

Dudley, R. M.
1967. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330.

Eberhart, R. and J. Kennedy
1995. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43. IEEE.

Finck, S., N. Hansen, R. Ros, and A. Auger
2010. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical report.

Finkel, D. E. and C. Kelley
2004. Convergence analysis of the direct algorithm. *Optimization On-line Digest.*

Freitas, N. D., M. Zoghi, and A. J. Smola
2012. Exponential regret bounds for gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Conference on Machine Learning*, J. Langford and J. Pineau, eds., pp. 1743–1750, New York, NY, USA. ACM.

Gaillard, P. and S. Gerchinovitz
2015. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory*, pp. 764–796.

Glynn, P. W. and D. L. Iglehart
1989. Importance sampling for stochastic simulations. *Management Science*, 35(11):1367–1392.

Grill, J.-B., M. Valko, and R. Munos
2015. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems*, pp. 667–675.

Grünewälder, S., J.-Y. Audibert, M. Opper, and J. Shawe-Taylor
2010. Regret bounds for gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 273–280.

Gutmann, H.-M.
2001. A radial basis function method for global optimization. *Journal of global optimization*, 19(3):201–227.

Hanneke, S.
2011. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361.

Hansen, N.
2006. The cma evolution strategy: a comparing review. In *Towards a New Evolutionary Computation*, pp. 75–102. Springer.

Hansen, N.
2011. The cma evolution strategy: A tutorial. Retrieved May 15, 2016, from http://www.lri.fr/hansen/cmaesintro.html.

Hansen, P., B. Jaumard, and S.-H. Lu
1992. Global optimization of univariate lipschitz functions: I. survey and properties. *Mathematical Programming*, 55(1-3):251–272.

Helton, J. C. and F. J. Davis
2003. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69.

Hennig, P. and C. J. Schuler
2012. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837.

Holland, J. H.
1975. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*.

Hooke, R. and T. A. Jeeves
1961. "direct search"solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229.

Huyer, W. and A. Neumaier
1999. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355.

Jamil, M. and X.-S. Yang
2013. A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194.

Johnson, D. S.
1973. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pp. 38–49. ACM.

Johnson, S. G.
2014. The NLopt nonlinear-optimization package. Retrieved May 15, 2016, from http://ab-initio.mit.edu/nlopt.

Jones, D. R., C. D. Perttunen, and B. E. Stuckman
1993. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181.

Jones, D. R., M. Schonlau, and W. J. Welch
1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Kaelo, P. and M. Ali
2006. Some variants of the controlled random search algorithm for global optimization. *Journal of Optimization Theory and Applications*, 130(2):253–264.

Kan, A. R. and G. T. Timmer
1987. Stochastic global optimization methods part i: Clustering methods. *Mathematical Programming*, 39(1):27–56.

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi, et al.
1983. Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kolda, T. G., R. M. Lewis, and V. Torczon
2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482.

Krause, A. and C. S. Ong
2011. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pp. 2447–2455.

Kushner, H. J.
1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Lewis, R. M. and V. Torczon
2002. A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089.

Lichman, M.
2013. UCI machine learning repository.

Malherbe, C., E. Contal, and N. Vayatis
2016. A ranking approach to global optimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48, pp. 1539–1547. PMLR.

Malherbe, C. and N. Vayatis
2016. A ranking approach to global optimization. *arXiv preprint arXiv:1603.04381*.

Malherbe, C. and N. Vayatis
2017. Global optimization of lipschitz functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2314–2323. PMLR.

Martinez-Cantin, R.
2014. Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research*, 15(1):3735–3739.

Matheron, G.
1963. Principles of geostatistics. *Economic geology*, 58(8):1246–1266.

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

Mladineo, R. H.
1986. An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200.

Močkus, J.
1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pp. 400–404. Springer.

Mockus, J.
1989. *Bayesian Approach to Global Optimization*, Mathematics and its Applications. Springer Netherlands.

Munos, R.
2014. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129.

Nelder, J. A. and R. Mead
1965. A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Niederreiter, H.
1988. Low-discrepancy and low-dispersion sequences. *Journal of number theory*, 30(1):51–70.

Pardalos, P. M., H. E. Romeijn, and H. Tuy
2000. Recent developments and trends in global optimization. *Journal of computational and Applied Mathematics*, 124(1):209–228.

Pintér, J. D.
1991. Global optimization in action. *Scientific American*, 264:54–63.

Piyavskii, S.
1972. An algorithm for finding the absolute extremum of a function. *USSR Computational Mathematics and Mathematical Physics*, 12(4):57–67.

Pollard, D.
1990. Empirical processes: theory and applications. In *NSF-CBMS regional conference series in probability and statistics*, pp. i–86. JSTOR.

Powell, M.
1998. Recent research at cambridge on radial basis functions. *New Developments in Approximation Theory*, p. 215.

Powell, M. J.
1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pp. 51–67. Springer.

Preux, P., R. Munos, and M. Valko
2014. Bandits attack function optimization. In *IEEE Congress on Evolutionary Computation*, pp. 2245–2252. IEEE.

Rasmussen, C. E. and C. K. Williams
2006. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Raz, R. and S. Safra
1997. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the 29th annual ACM symposium on Theory of computing*, pp. 475–484. ACM.

Reaume, D. J., H. E. Romeijn, and R. L. Smith
2001. Implementing pure adaptive search for global optimization using markov chain sampling. *Journal of Global Optimization*, 20(1):33–47.

Rechenberg, I.
1971. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologishen Evolution.* PhD thesis, Technical University of Berlin.

Rios, L. M. and N. V. Sahinidis
2013. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293.

Roberts, S. J.
2010. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature.* PhD thesis, University of Oxford.

Robertson, B., C. Price, and M. Reale
2013. Cartopt: a random search method for nonsmooth unconstrained optimization. *Computational Optimization and Applications*, 56(2):291–315.

Roth, A., T. Dreyfus, C. H. Robert, and F. Cazals
2015. *Hybridizing rapidly growing random trees and basin hopping yields an improved exploration of energy landscapes.* PhD thesis, Inria.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn
1989. Design and analysis of computer experiments. *Statistical science*, pp. 409–423.

Sarkar, D., E. Contal, N. Vayatis, and F. Dias
2016. Prediction and optimization of wave energy converter arrays using a machine learning approach. *Renewable Energy*, 97:504–517.

Sergeyev, Y. D., R. G. Strongin, and D. Lera
2013. *Introduction to global optimization exploiting space-filling curves*. Springer Science & Business Media.

Shubert, B. O.
1972. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388.

Smith, R. L.
1984. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308.

Snoek, J., H. Larochelle, and R. P. Adams
2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959.

Sobol, I. and Y. L. Levitan
1976. The production of points uniformly distributed in a multidimensional cube. *Preprint IPM Akad. Nauk SSSR*, 40(3).

Sobol, I. M.
1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802.

Spendley, W., G. R. Hext, and F. R. Himsworth
1962. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4):441–461.

Srinivas, N., A. Krause, S. M. Kakade, and M. W. Seeger
2012. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

Srinivas, N., A. Krause, M. Seeger, and S. M. Kakade
2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, J. Fürnkranz and T. Joachims, eds., pp. 1015–1022. Omnipress.

Stein, M.
1987. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151.

Surjanovic, S. and D. Bingham
2013. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 15, 2016, from http://www.sfu.ca/~ssurjano.

Talagrand, M.
2014. *Upper and lower bounds for stochastic processes: modern methods and classical problems*, volume 60. Springer Science & Business Media.

Teytaud, O. and H. Fournier
2008. Lower bounds for evolution strategies using vc-dimension. In *International Conference on Parallel Problem Solving from Nature*, pp. 102–111. Springer.

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

Tsybakov, A. B.
1997. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969.

Valko, M., A. Carpentier, and R. Munos
2013. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 19–27.

Vapnik, V.
1992. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838.

Vazquez, E. and J. Bect
2010. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095.

Walsh, G.
1970. *Methods of optimization*. John Wiley & Sons Ltd.

Wang, Z., C. Li, S. Jegelka, and P. Kohli
2017. Batched high-dimensional bayesian optimization via structural kernel learning. *arXiv preprint arXiv:1703.01973*.

Wang, Z., M. Zoghi, F. Hutter, D. Matheson, N. De Freitas, et al.
2013. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conference on Artificial Intelligence*, pp. 1778–1784.

Willett, R. M. and R. D. Nowak
2007. Minimax optimal level-set estimation. *IEEE Transactions on Image Processing*, 16(12):2965–2979.

Winfield, D. H.
1970. *Function and functional optimization by interpolation in data tables*. PhD thesis, Harvard University.

Wood, G. and B. Zhang
1996. Estimation of the lipschitz constant of a function. *Journal of Global Optimization*, 8(1):91–103.

Yu, Y., H. Qian, and Y.-Q. Hu
2016. Derivative-free optimization via classification. In *AAAI*, pp. 2286–2292.

Zabinsky, Z. B.
2003. *Stochastic adaptive search for global optimization*, volume 72. Springer Science & Business Media.

Zabinsky, Z. B. and R. L. Smith
1992. Pure adaptive search in global optimization. *Mathematical Programming*, 53(1-3):323–338.

Zabinsky, Z. B., R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman
1993. Improving hit-and-run for global optimization. *Journal of Global Optimization*, 3(2):171–192.

Zhigljavsky, A. and J. Pintér
1991. *Theory of Global Random Search*, Mathematics and its Applications. Springer Netherlands.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Quelques contributions à l'optimisation globale

Mots clés : Optimisation, fonctions lipschitziennes, ordonnancement binaire, analyse statistique

Résumé : Ce travail de thèse s'intéresse au problème d'optimisation séquentielle d'une fonction inconnue définie sur un ensemble continu et borné. Ce type de problème apparaît notamment dans la conception de systèmes complexes, lorsque l'on cherche à optimiser le résultat de simulations numériques ou plus simplement lorsque la fonction que l'on souhaite optimiser ne présente aucune forme de régularité évidente comme la linéarité ou la convexité. Dans un premier temps, nous nous focalisons sur le cas particulier des fonctions lipschitziennes. Nous introduisons deux nouvelles stratégies ayant pour but d'optimiser n'importe quelle fonction de coefficient de Lipschitz connu puis inconnu. Ensuite, en introduisant différentes mesures de régularité, nous formulons et obtenons des résultats de consistance pour ces méthodes ainsi que des vitesses de convergence sur leurs erreurs d'approximation.

Dans une seconde partie, nous nous proposons d'explorer le domaine de l'ordonnancement binaire dans le but de développer des stratégies d'optimisation pour fonctions non régulières. En observant que l'apprentissage de la règle d'ordonnancement induite par la fonction inconue permet l'identification systématique de son optimum, nous faisons le lien entre théorie de l'ordonnacement et théorie de l'optimisation, ce qui nous permet de développer de nouvelles méthodes reposant sur le choix de n'importe quelle technique d'ordonnancement et de formuler différents résultats de convergence pour l'optimisation de fonctions non régulières. Enfin, les stratégies d'optimisation développées au cours de la thèse sont comparées aux méthodes présentes dans l'état de l'art sur des problèmes de calibration de systèmes d'apprentissages ainsi que sur des problèmes synthétiques fréquemment rencontrés dans le domaine de l'optimisation globale.

Title : Global optimization: contributions

Keywords : Optimization, Lipschitz constant, ranking, statistical analysis

Abstract : This work addresses the sequential optimization of an unknown and potentially non-convex function over a continuous and bounded set. These problems are of particular interest when evaluating the function requires numerical simulations with significant computational cost or when the objective function does not satisfy the standard properties used in optimization such as linearity or convexity. In a first part, we consider the problem of designing sequential strategies which lead to efficient optimization of an unknown function under the only assumption that it has finite Lipschitz constant. We introduce and analyze two strategies which aim at optimizing any function with fixed and unknown Lipschitz constant. Consistency and minimax rates for these algorithms are proved, as well as fast rates under an additional Hölder like condition.

In a second part, we propose to explore concepts from ranking theory based on overlaying level sets in order to develop optimization methods that do not rely on the smoothness of the function. We observe that the optimization of the function essentially relies on learning the bipartite rule it induces. Based on this idea, we relate global optimization to bipartite ranking which allows to address the cases of functions with weak regularity properties. Novel meta algorithms for global optimization which rely on the choice of any bipartite ranking method are introduced and theoretical properties are provided in terms of statistical consistency and finite-time convergence toward the optimum. Eventually, the algorithms developed in the thesis are compared to existing state-of-the-art methods over typical benchmark problems for global optimization.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France