Global optimization of Lipschitz functions

Cédric Malherbe [1] **Nicolas Vayatis** [1]

Abstract

The goal of the paper 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. Similar theoretical guarantees are shown to hold for the adaptive algorithm and a numerical assessment is provided at the end of the paper to illustrate the potential of this strategy with respect to state-of-the-art methods over typical benchmark problems for global optimization.

1. Introduction

In many applications such as complex system design or hyperparameter calibration for learning systems, the goal is to optimize the output value of an unknown function with as few evaluations as possible. Indeed, in such contexts, evaluating the performance of a single set of parameters often requires numerical simulations or cross-validations with significant computational cost and the operational constraints impose a sequential exploration of the solution space with small samples. Moreover, it can generally not be assumed that the function has good properties such as linearity or convexity. This generic problem of sequentially optimizing the output of an unknown and potentially nonconvex function is often referred to as global optimization (Pintér, 1991), black-box optimization (Jones et al., 1998) or derivative-free optimization (Rios & Sahinidis, 2013). There is a large number of algorithms based on various heuristics which have been introduced in order to solve this problem such as genetic algorithms, model-based methods or Bayesian optimization. We focus here 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; 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 & Neumaier, 1999) or SOO (Preux et al., 2014). Convergence properties of global optimization methods have been developed in the works of (Valko et al., 2013; 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 function and display good performance in typical benchmarks for optimization. Second, we show that these algorithms can 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 paper is organized as follows. In Section 2, we introduce the framework and give generic results about the convergence of sequential algorithms. In Section 3, we introduce and analyze the LIPO algorithm which requires the knowledge of the Lipschitz constant. In Section 4, the algorithm is extended to the case where the Lipschitz constant is unknown and the adaptive algorithm is compared to existing methods in Section 5. All proofs can be found in the Supplementary Material provided as a separate document.

[1]CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235, Cachan, France. Correspondence to: <name@cmla.ens-cachan.fr>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

2. Setup and preliminary results

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 $\mathcal{X}$. 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 $f(X_1)$ at an initial point X_1 and then selects at each step $t \geq 1$ an evaluation point $X_{t+1} \in \mathcal{X}$ depending on the previous evaluations $(X_1, f(X_1)), \ldots, (X_t, f(X_t))$ and receives 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 $\hat{\imath}_n \in \arg\min_{i=1\ldots n} f(X_i)$ 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 highest evaluation observed so far:

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

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. Moreover, the assumption made on the unknown function f throughout the paper is that it has a finite Lipschitz constant k, *i.e.*

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

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

Notations. For all $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we denote by $\|x\|_2 = (\sum_{i=1}^d x_i^2)^{1/2}$ 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 $\text{rad}(\mathcal{X}) = \max\{r > 0 : \exists x \in \mathcal{X} \text{ such that } B(x, r) \subseteq \mathcal{X}\}$, its diameter as $\text{diam}(\mathcal{X}) = \max_{(x,x') \in \mathcal{X}^2} \|x - x'\|_2$ and we denote by $\mu(\mathcal{X})$ its volume where $\mu(\cdot)$ stands for the Lebesgue measure. $\text{Lip}(k) = \{f : \mathcal{X} \to \mathbb{R} \text{ s.t. } |f(x) - f(x')| \leq k \cdot \|x - x'\|_2, \ \forall (x, x') \in \mathcal{X}^2\}$ denotes the class of k-Lipschitz functions defined on $\mathcal{X}$ and $\bigcup_{k \geq 0} \text{Lip}(k)$ denotes the set of Lipschitz continuous functions. $\mathcal{U}(\mathcal{X})$ 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. Preliminary results

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

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

Definition 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 distributions $\{A_t\}_{t \geq 2}$ taking values in $\mathcal{X}$ and based on the previous evaluations which define the iteration loop, i.e. $X_{t+1} \sim A_{t+1}((X_1, f(X_1)), \ldots, (X_t, f(X_t)))$.*

Note that this class of algorithms also includes the 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 (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 $\mathcal{X}$ if and only if*

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

where $X_1, \ldots, X_n$ denotes a sequence of n 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 3 (CONSISTENCY NSC) *A global optimization algorithm A is consistent over the set of Lipschitz functions if and only if*

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

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 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 with probability at least $1-\delta$ and independently of the function values,*

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

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 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 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 $\tilde{f} \in \bigcup_{k\geq 0} \operatorname{Lip}(k)$ 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 however 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 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:*

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

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

We point out that this minimax rate of convergence 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 $f \in \operatorname{Lip}(k)$, then an optimal covering rate necessarily implies minimax efficiency. However, as it can be seen by examining the proof of Proposition 6 provided in the Supplementary Material, 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 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 4).

3. Optimization with fixed Lipschitz constant

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

3.1. The LIPO Algorithm

The inputs of the LIPO algorithm (**Algorithm 1**) are a number n of function evaluations, a Lipschitz constant $k \geq 0$, the input space $\mathcal{X}$ and the unknown function f. 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 upper bound on possible values $UB : x \mapsto \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2$ evaluated at this point and computed from the previous evaluations is at least equal to the value of the best evaluation observed so far $\max_{i=1\dots t} f(X_i)$. As an example, the computation of the decision rule of LIPO is illustrated in Figure 1.

Algorithm 1 LIPO$(n, k, \mathcal{X}, f)$

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 $\min_{i=1\dots t} (f(X_i) + k \cdot \|X_{t+1} - X_i\|_2) \geq \max_{i=1\dots t} f(X_i)$
Evaluate $f(X_{t+1})$, $t \leftarrow t+1$

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

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

Definition 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:*

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

Indeed, one can recover from this definition the subset of points which can actually maximize the function f.

Definition 8 (POTENTIAL MAXIMIZERS) *Using the same notations as in Definition 7, we define the subset of potential maximizers estimated over any sample $t \geq 1$ evaluations with a constant $k \geq 0$ as follows:*

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

We may now provide an equivalence which makes the link with the decision rule of the LIPO algorithm.

Lemma 9 *If $\mathcal{X}_{k,t}$ denotes the set of potential maximizers defined above, then we have the following equivalence:*

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

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

Remark 10 (EXTENSION TO OTHER SMOOTHNESS ASSUMPTIONS) *It is important to note the proposed optimization scheme could easily be extended to a large number of sets 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 locally smooth around their maxima with regards to any semi-metric $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ previously considered in (Munos, 2014), a straightforward derivation of Lemma 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 will only derive convergence results for the version of the algorithm stated above.*

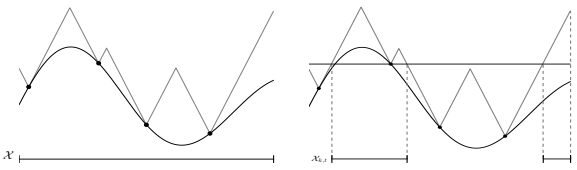

Figure 1. Left: A Lipschitz function, a sample of 4 evaluations and the upper bound $UB : x \mapsto \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|_2$ in grey. *Right:* the set of points $\mathcal{X}_{k,t} := \{x \in \mathcal{X} : UB(x) \geq \max_{i=1\dots t} f(X_i)\}$ which satisfy the decision rule.

3.2. Convergence analysis

We start with the consistency property of the algorithm.

Proposition 11 (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.,*

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

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 PRS.

Proposition 12 (FASTER THAN PURE RANDOM SEARCH) *Consider the LIPO algorithm tuned with any constant $k \geq 0$. Then, for any $f \in \text{Lip}(k)$ 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 13 (UPPER BOUND) *For any $f \in \text{Lip}(k)$, $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, we have with probability at least $1 - \delta$,*

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

This bound which assesses the miminax optimality of LIPO stated in Proposition 6 does however not show any improvement over PRS and it cannot be significantly improved without any additional assumption as shown below.

Proposition 14 *For any $n \in \mathbb{N}^\star$ and $\delta \in (0,1)$, there exists a function $\tilde{f} \in \text{Lip}(k)$ only depending on n and δ for which we have with probability at least $1 - \delta$:*

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

As announced in Section 2.2, 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.

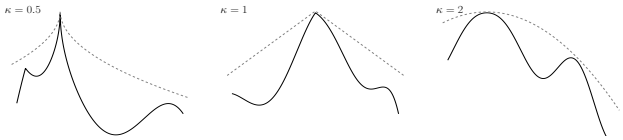

Figure 2. Three one-dimensional functions satisfying Condition 1 with $\kappa = 1/2$ (*Left*), $\kappa = 1$ (*Middle*) and $\kappa = 2$ (*Right*).

Condition 1 (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 & 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 which 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 3.2.

Theorem 15 (FAST RATES) *Let $f \in \mathrm{Lip}(k)$ be any Lipschitz function satisfying Condition 1 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 $C_{k,\kappa} = (c_\kappa \max_{x \in \mathcal{X}} \|x - x^\star\|^{\kappa-1} / 8k)^d$.

The last result we provide states an exponentially decreasing lower bound.

Theorem 16 (LOWER BOUND) *For any $f \in \mathrm{Lip}(k)$ satisfying Condition 1 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 $1 - \delta$,*

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

A discussion on these results can be found in the next section where LIPO is compared with similar algorithms.

3.3. Comparison with previous works

The Piyavskii algorithm (Piyavskii, 1972) is a Lipschitz method with fixed $k \geq 0$ consisting in sequentially evaluating the function over a point $X_{t+1} \in \arg\max_{x \in \mathcal{X}} \min_{i=1\dots t} f(X_i) + k \cdot \|x - X_i\|$ maximizing the upper bound displayed on Figure 1. (Munos, 2014) also proposed a similar algorithm (DOO) which uses a hierarchical partitioning of the space in order to sequentially expand and evaluate the function over the center of a partition which has the highest upper bound computed from a semi-metric ℓ set as input. Up to our knowledge, only the consistency of the Piyavskii algorithm was proven in (Mladineo, 1986) and (Munos, 2014) derived finite-time bounds for DOO with the use of weaker local assumptions. To compare our results, we thus considered DOO tuned with $\ell(x, x') = k \|x - x'\|_2$ over $\mathcal{X} = [0,1]^d$ partitioned into a 2^d-ary tree of hypercubes and with f belonging to the sets of globally smooth functions: (a) $\mathrm{Lip}(k)$, (b) $\mathcal{F}_\kappa = \{f \in \mathrm{Lip}(k) \text{ satisfying Condition 1 with } c_\kappa, \kappa \geq 0\}$ 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 1. In addition 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 a similar exponenital rate, up to a constant factor, when $\kappa = 1$. Hence, when f is only known to be k-Lipschitz, 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 are not necessarily optimal, it is also interesting to note that the term $(k\sqrt{d}/c_\kappa)^d$ appearing in both the exponential 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 one should expect an algorithm exploiting the local smoothness k_ℓ 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 1. Comparison of the results reported over the difference $\max_{x \in \mathcal{X}} f(x) - \max_{i=1\dots n} f(X_i)$ in Lipschitz optimization. Dash symbols are used when no results could be found.

4. Optimization with unknown Lipschitz constant

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

4.1. The adaptive algorithm

The AdaLIPO algorithm (**Algorithm 2**) is an extension of LIPO which involves an estimate of the Lipschitz constant and takes 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 alternates randomly between two distinct phases: exploration and exploitation. Indeed, at step $t < n$, a Bernoulli random variable B_{t+1} of parameter p driving this trade-off is sampled. If $B_{t+1} = 1$, then 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 $\hat{k}_t$ which is 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 17 (EXAMPLES OF MESHGRIDS) *Several sequences of Lipschitz constants with various shapes such as $k_i = |i|^{sgn(i)}$, $\ln(1+|i|^{sgn(i)})$ or $(1+\alpha)^i$ for some $\alpha > 0$ could be considered to implement the algorithm. In particular, we point out that with these sequences the computation of the estimate is straightforward. For instance, when $k_i = (1+\alpha)^i$, we have $\hat{k}_t = (1+\alpha)^{i_t}$ where $i_t = \lceil \ln(\max_{i\neq j} |f(X_j) - f(X_l)|/\|X_j - X_l\|_2)/\ln(1+\alpha) \rceil$.*

4.2. Convergence analysis

Lipschitz constant estimate. Before starting the analysis of AdaLIPO, we first provide a control on the Lipschitz constant estimate based on a sample of random evaluations that will be useful to analyse its performance. In particular, the next result illustrates the purpose of using a discretization of Lipschitz constant 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 18 *Let f 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*

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

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

Algorithm 2 ADALIPO$(n, p, k_{i\in\mathbb{Z}}, \mathcal{X}, f)$

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 introduced in Definition 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 $\hat{\imath}_n \in \arg\max_{i=1\dots n} f(X_i)$

Remark 19 (MEASURE OF GLOBAL SMOOTHNESS) *The coefficient $\Gamma(f, k_{i^\star-1})$ which appears in the lower bound of Proposition 18 can be seen as a measure of the global smoothness of the function f with regards to $k_{i^\star-1}$. Indeed, observing that $1/\lfloor t/2 \rfloor \sum_{i=1}^{\lfloor t/2 \rfloor} \mathbb{I}\{|f(X_i) - f(X_{i+\lfloor t/2 \rfloor})| > k_{i^\star-1}\|X_i - X_{\lfloor t/2 \rfloor+i}\|_2\} \xrightarrow{p} \Gamma(f, k_{i^\star-1})$, it is easy to see that Γ 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 20 (DENSITY OF THE SEQUENCE) *As a direct consequence of the previous remark, we point out that the density of the sequence $k_{i\in\mathbb{Z}}$, captured here by $\alpha = \sup_{i\in\mathbb{Z}}(k_{i+1} - k_i)/k_i$ has opposite impacts on the maximal deviation of the estimate and its convergence rate. Indeed, since α is involved in both the following upper bounds on the deviation $(\lim_{t\to\infty} \hat{k}_t - k^\star)/k^\star \leq \alpha$ where $k^\star = \sup\{k \geq 0 : f \notin \text{Lip}(k)\}$ and on the coefficient $\Gamma(f, k_{i^\star-1}) \leq \Gamma(f, k^\star/(1+\alpha))$, we deduce that using a sequence with a small α reduces the bias but also the convergence rate through a small coefficient $\Gamma(f, k_{i^\star-1})$.*

Analysis of AdaLIPO. Given the consistency equivalence of Proposition 3, one can directly obtain the following asymptotic result.

Proposition 21 (CONSISTENCY) *The AdaLIPO algorithm tuned with any parameter $p \in (0,1)$ and any sequence of*

Lipschitz constant $k_{i\in\mathbb{Z}}$ covering $\mathbb{R}^+$ 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 22 (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 $1-\delta$,*

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

where $\Gamma(f,k_{i^\star-1})$ and $i^\star$ are defined as in Proposition 18.

This result might be misleading since it advocates that doing pure exploration gives the best rate (*i.e.,* when $p \to 1$). However, as Proposition 18 provides us with the guarantee that $f \in \mathrm{Lip}(\hat{k}_t)$ within a finite number of iterations where $\hat{k}_t$ denotes the Lipschitz constant estimate, one can recover faster convergence rates similar to the one reported for LIPO where the constant k is assumed to be known.

Theorem 23 (FAST RATES) *Consider the same assumptions as in Proposition 22 and assume in addition that the function f satisfies Condition 1 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 $1-\delta$,*

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

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. In particular, 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.

4.3. Comparison with previous works

The DIRECT algorithm (Jones et al., 1993) is a Lipschitz algorithm with unknown constant which uses a deterministic splitting technique of the search space to evaluate the function on subdivisions of the space that have recorded the highest evaluation among all subdivisions of similar size. Moreover, (Munos, 2014) generalized DIRECT in a broader setting by extending DOO to any unknown and arbitrary local semi-metric. 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 smooth functions. However, as far as we know, only the consistency property of DIRECT was shown in (Finkel & Kelley, 2004) and (Munos, 2014) derived convergence rates of the same order as for DOO, except that the best rate they derive is of order $O(e^{-c\sqrt{n}})$ to be compared with the fast rate of AdaLIPO which is of order $O^*_{\mathbb{P}}(e^{-cn})$. The conclusion of the comparison thus remains the same as in Section 3: exploiting the global smoothness instead of just the local one allows to derive faster algorithms in the some cases where the unknown function is indeed globally smooth.

5. Experiments

We compare here the empirical performance of AdaLIPO with five state-of-the-art global optimization methods.

Algorithms. BAYESOPT* (Martinez-Cantin, 2014) is a Bayesian optimization algorithm which uses a distribution over functions to build a surrogate model of the unknown function. The parameters of the distribution are estimated during the optimization process. CMA-ES‡(Hansen, 2006) is an evolutionary algorithm which samples the new evaluation points according to a multivariate normal distribution with mean vector and covariance matrix computed from the previous evaluations. CRS†(Kaelo & Ali, 2006) is a variant of PRS including local mutations which starts with a random population and evolves these points by an heuristic rule. MLSL†(Kan & Timmer, 1987) is a multistart algorithm performing 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. DIRECT†(Jones et al., 1993) and PRS were previously introduced. For a fair comparison, the tuning parameters were all set to default and AdaLIPO was constantly used with a parameter p set to 0.1 and a sequence $k_i = (1+0.01/d)^i$ fixed by an arbitrary rule of thumb. [1]

Data sets. Following the steps of (Malherbe & Vayatis, 2016), we first studied the task of estimating the regularization parameter λ and the bandwidth σ of a gaussian kernel ridge regression minimizing the empirical mean squared

[1] In Python 2.7 from *BayesOpt (Martinez-Cantin, 2014), ‡CMA 1.1.06 (Hansen, 2011) and †NLOpt (Johnson, 2014).

Problem	Auto-MPG	BreastCancer	Concrete	Housing	Yacht	HolderTable	Rosenbrock	LinearSlope	Sphere	Deb N.1	
AdaLIPO	14.6 (±09)	**05.4** (±03)	**04.9** (±02)	**05.4** (±04)	25.2 (±21)	**077** (±058)	07.5 (±07)	029 (±13)	036 (±12)	916(±225)	
BayesOpt	**10.8** (±03)	06.8 (±04)	06.4 (±03)	07.5 (±04)	13.8 (±20)	410 (±417)	07.6 (±05)	032 (±58)	019 (±03)	814(±276)	
CMA-ES	29.3 (±25)	11.1 (±09)	10.4 (±08)	12.4 (±12)	29.6 (±25)	080 (±115)	10.0 (±10)	100 (±76)	171 (±68)	930(±166)	
CRS	28.7 (±14)	08.9 (±08)	10.0 (±09)	13.8 (±10)	32.6 (±15)	307 (±422)	09.0 (±09)	094 (±43)	233 (±54)	980(±166)	
DIRECT	11.0 (±00)	06.0 (±00)	06.0 (±00)	06.0 (±00)	**11.0** (±00)	080 (±000)	10.0 (±00)	092 (±00)	**031** (±00)	1000(±00)	
MLSL	13.1 (±15)	06.6 (±03)	06.1 (±04)	07.2 (±03)	14.4 (±13)	305 (±379)	**06.9** (±05)	**016** (±33)	175(±302)	**198**(±326)	
PRS	65.1 (±62)	10.6 (±10)	09.8 (±09)	11.5 (±10)	73.3 (±72)	210 (±202)	09.0 (±09)	831(±283)	924(±210)	977(±117)	**target 90%**
AdaLIPO	17.7 (±09)	**06.6** (±04)	**06.4** (±04)	17.9 (±25)	33.3 (±26)	102 (±065)	11.5 (±11)	053 (±22)	**042** (±11)	986(±255)	
BayesOpt	12.2 (±06)	08.4 (±03)	07.9 (±03)	**13.9** (±22)	**15.9** (±21)	418 (±410)	12.0 (±08)	032 (±59)	045 (±16)	949(±153)	
CMA-ES	42.9 (±31)	13.7 (±10)	13.5 (±10)	23.0 (±16)	40.5 (±30)	136 (±184)	16.1 (±13)	151 (±94)	223 (±57)	952(±127)	
CRS	35.8 (±13)	13.6 (±10)	14.6 (±11)	22.8 (±12)	38.3 (±31)	580 (±444)	15.8 (±14)	131 (±62)	340 (±66)	997(±127)	
DIRECT	**11.0** (±00)	11.0 (±00)	11.0 (±00)	19.0 (±00)	27.0 (±00)	**080** (±000)	10.0 (±00)	116 (±00)	098 (±00)	1000(±00)	
MLSL	15.0 (±15)	07.6 (±03)	07.3 (±04)	16.3 (±10)	16.3 (±13)	316 (±384)	**08.8** (±05)	**018** (±37)	226(±336)	**215**(±328)	
PRS	139 (±131)	17.7 (±17)	14.0 (±12)	39.6 (±39)	247(±249)	349 (±290)	18.0 (±17)	985(±104)	1000(±00)	998(±025)	**target 95%**
AdaLIPO	32.6 (±16)	34.1 (±36)	70.8 (±58)	65.4 (±62)	61.7 (±39)	212 (±129)	44.6 (±39)	122 (±31)	**052** (±10)	1000(±00)	
BayesOpt	**14.0** (±07)	31.0 (±51)	28.2 (±34)	17.9 (±22)	**18.5** (±22)	422 (±407)	27.6 (±22)	032 (±59)	222 (±77)	1000(±00)	
CMA-ES	73.7 (±49)	35.1 (±20)	46.3 (±29)	61.5 (±85)	70.9 (±50)	215 (±198)	43.5 (±37)	211 (±92)	308 (±60)	962(±106)	
CRS	48.5 (±16)	34.8 (±12)	36.6 (±15)	43.7 (±14)	52.9 (±18)	599 (±427)	42.7 (±23)	168 (±76)	607 (±81)	1000(±00)	
DIRECT	47.0 (±00)	27.0 (±00)	37.0 (±00)	41.0 (±00)	49.0 (±00)	**080** (±000)	24.0 (±00)	226 (±00)	548 (±00)	1000(±00)	
MLSL	20.6 (±17)	**12.8** (±03)	**14.7** (±10)	**16.3** (±10)	21.4 (±14)	322 (±382)	**19.4** (±49)	**022** (±42)	304(±357)	**256**(±334)	
PRS	747(±330)	145(±124)	176(±148)	406(±312)	779(±334)	772 (±310)	100(±106)	1000(±00)	1000(±00)	1000(±00)	**target 99%**

Table 2. Results of the numerical experiments. The table displays the number of evaluations required by each method to reach the specified target (mean ± standard deviation). In bold, the best result obtained in terms of average of function evaluations.

error of the predictions over a 10-fold cross validation with real data sets. The optimization was performed over $(\ln(\lambda), \ln(\sigma)) \in [-3, 5] \times [-2, 2]$ with five data sets from the UCI Machine Learning Repository (Lichman, 2013): *Auto-MPG*, *Breast Cancer Wisconsin (Prognostic)*, *Concrete slump test*, *Housing* and *Yacht Hydrodynamics*. We then compared the algorithms on a series of five synthetic problems commonly met in standard optimization benchmark taken from (Jamil & Yang, 2013; Surjanovic & Bingham, 2013): *HolderTable*, *Rosenbrock*, *Sphere*, *LinearSlope* and *Deb N.1*. This series includes multimodal and non-linear functions as well as ill-conditioned and well-shaped functions with a dimensionality ranging from 2 to 5. A complete description of the test functions of the benchmark can be found in the Supplementary Material.

Protocol and performance metrics. For each problem and each algorithm, 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 $\tau_k := \min\{i = 1, \dots, n : f(X_i^{(k)}) \geq f_{\text{target}}(t)\}$ where $\min\{\emptyset\} = 1000$ by convention, $\{f(X_i^{(k)})\}_{i=1}^n$ denotes the evaluations made by a given method on the k-th run with $k \leq K$ and the target value is set to $f_{\text{target}}(t) := \max_{x \in \mathcal{X}} f(x) - \big(\max_{x \in \mathcal{X}} f(x) - \int_{x \in \mathcal{X}} f(x)\, \mathrm{d}x / \mu(\mathcal{X})\big) \times (1 - t)$. The normalization of the target to the average value prevents the performance measures from being dependent of any constant term in the unknown function. In practice, the average was estimated from a Monte Carlo sampling of 10^6 evaluations and the maximum by taking the best value observed over all the sets of experiments. Based on these stopping times, we computed the average and standard deviation of the number of evaluations required to reach the target, *i.e.* $\bar{\tau}_K = \sum_{k=1}^K \tau_k / K$ and $\hat{\sigma}_\tau = (\sum_{k=1}^K (\tau_k - \bar{\tau}_K)^2 / K)^{1/2}$.

Results. Results are collected in Table 2. Due to space constraints, we only make few comments. First, we point out that the proposed method displays very competitive results over most of the problems of the benchmark (except on the non-smooth *DebN.1* where most methods fail). In particular, AdaLIPO obtains several times the best performance for the target 90% and 95% (see, *e.g.*, *BreastCancer, HolderTable, Sphere*) and experiments *Linear Slope* and *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, 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 around the maxima but could certainly be solved with the addition of a noise parameter that would allow the algorithm be more robust against local perturbations. Additionally, investigating better values for p and k_i as well as alternative covering methods such as LHS (Stein, 1987) could also be promising approaches to improve its performance. However, an empirical analysis of the algorithm with these extensions is beyond the scope of the paper and will be carried out in a future work.

6. Conclusion

We introduced two novel strategies for global optimization: LIPO which requires the knowledge of the Lipschitz constant and its adaptive version AdaLIPO which estimates the constant during the optimization process. A theoretical analysis is provided and empirical results based on synthetic and real problems have been obtained demonstrating the performance of the adaptive algorithm with regards to existing state-of-the-art global optimization methods.

References

Bull, Adam D. Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904, 2011.

Dasgupta, Sanjoy. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011.

Finkel, Daniel E and Kelley, CT. Convergence analysis of the direct algorithm. *Optimization On-line Digest*, 2004.

Grill, Jean-Bastien, Valko, Michal, and Munos, Rémi. Black-box optimization of noisy functions with unknown smoothness. In *Neural Information Processing Systems*, 2015.

Hanneke, Steve. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

Hansen, Nikolaus. The cma evolution strategy: a comparing review. In *Towards a New Evolutionary Computation*, pp. 75–102. Springer, 2006.

Hansen, Nikolaus. The cma evolution strategy: A tutorial. Retrieved May 15, 2016, from http://www.lri.fr/hansen/cmaesintro.html, 2011.

Huyer, Waltraud and Neumaier, Arnold. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.

Jamil, Momin and Yang, Xin-She. A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

Johnson, Steven G. The NLopt nonlinear-optimization package. Retrieved May 15, 2016, from http://ab-initio.mit.edu/nlopt, 2014.

Jones, Donald R, Perttunen, Cary D, and Stuckman, Bruce E. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

Jones, Donald R., Schonlau, Matthias, and Welch, William J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Kaelo, Professor and Ali, Montaz. Some variants of the controlled random search algorithm for global optimization. *Journal of Optimization Theory and Applications*, 130(2):253–264, 2006.

Kan, AHG Rinnooy and Timmer, Gerrit T. Stochastic global optimization methods part i: Clustering methods. *Mathematical Programming*, 39(1):27–56, 1987.

Lichman, Moshe. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Malherbe, Cédric and Vayatis, Nicolas. A ranking approach to global optimization. *arXiv preprint arXiv:1603.04381*, 2016.

Malherbe, Cédric, Contal, Emile, and Vayatis, Nicolas. A ranking approach to global optimization. In *In Proceedings of the 33st International Conference on Machine Learning*, pp. 1539–1547, 2016.

Martinez-Cantin, Ruben. Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research*, 15(1):3735–3739, 2014.

Mladineo, Regina Hunter. An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200, 1986.

Munos, Rémi. 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, 2014.

Pintér, János D. Global optimization in action. *Scientific American*, 264:54–63, 1991.

Piyavskii, SA. An algorithm for finding the absolute extremum of a function. *USSR Computational Mathematics and Mathematical Physics*, 12(4):57–67, 1972.

Preux, Philippe, Munos, Rémi, and Valko, Michal. Bandits attack function optimization. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 2245–2252. IEEE, 2014.

Rios, Luis Miguel and Sahinidis, Nikolaos V. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

Shubert, Bruno O. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

Stein, Michael. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2): 143–151, 1987.

Surjanovic, Sonja and Bingham, Derek. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 15, 2016, from http://www.sfu.ca/~ssurjano, 2013.

Valko, Michal, Carpentier, Alexandra, and Munos, Rémi. Stochastic simultaneous optimistic optimization. In *In Proceedings of the 30th International Conference on Machine Learning*, pp. 19–27, 2013.

Zhigljavsky, A.A. and Pintér, J.D. *Theory of Global Random Search*. Mathematics and its Applications. Springer Netherlands, 1991. ISBN 9780792311225.